



Silk Central 20.6

[Help](#)

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
<http://www.microfocus.com>

© Copyright 2004-2020 Micro Focus or one of its affiliates.

MICRO FOCUS, the Micro Focus logo and Silk Central are trademarks or registered trademarks of Micro Focus or one of its affiliates.

All other marks are the property of their respective owners.

2020-05-26

Contents

| | |
|---|-----------|
| Silk Central Help | 9 |
| What's New in Silk Central 20.6 | 10 |
| Administration | 10 |
| Dashboard | 10 |
| Requirements | 10 |
| Tests | 11 |
| Execution Planning | 11 |
| Issue Manager | 12 |
| Manual Testing | 12 |
| Executions | 12 |
| Integrations | 12 |
| Microsoft SQL Server Support | 13 |
| Getting Started | 14 |
| Installing and Licensing Silk Central | 14 |
| Silk Central Concepts | 14 |
| Workflow | 14 |
| Requirements | 15 |
| Tests | 15 |
| Execution Planning | 16 |
| User Interface | 16 |
| Tour of the UI | 16 |
| Dashboard | 19 |
| Browser Settings | 24 |
| Login and Logout | 24 |
| Accessibility | 24 |
| Quick Start Tasks | 25 |
| Configuring Required Assets - Quick Start Task | 25 |
| Defining Requirements - Quick Start Task | 26 |
| Designing Tests - Quick Start Task | 28 |
| Execution Planning and Executing Tests - Quick Start Task | 29 |
| Tracking and Analyzing Test Results - Quick Start Task | 34 |
| Requirements | 36 |
| Configuring Requirement Settings and Integrations | 36 |
| Requirement Properties | 36 |
| Requirements Management Tools Integration | 41 |
| Filters | 56 |
| Change Notification | 60 |
| Managing Requirements | 61 |
| Creating Requirements | 62 |
| Editing Requirements | 62 |
| Marking Requirements as Obsolete | 63 |
| Recovering Obsolete Requirements | 63 |
| Permanently Deleting Obsolete Requirements | 63 |
| Requirements Reports | 64 |
| Coverage | 65 |
| Flags | 66 |
| Configuring Requirement Properties | 68 |
| Replacing Requirement Properties | 68 |
| Finding Requirement Properties | 69 |
| Requirement Properties Page | 69 |

| | |
|---|-----------|
| Requirement Attachments | 70 |
| Associating Tests with Requirements | 72 |
| Assigning Tests from Grid View to Requirements | 72 |
| Assigning Tests to Requirements Manually | 73 |
| Locating Assigned Tests in the Tests Tree | 73 |
| Removing Test Assignments | 73 |
| Generating Tests from Requirements Details View | 74 |
| Assigned Tests Page | 74 |
| Requirements Import | 76 |
| Importing Requirements from a Word File | 76 |
| Updating Requirements from a Word File | 77 |
| Word Requirements File | 77 |
| Importing Requirements from an Excel File | 78 |
| Updating Requirements from an Excel File | 79 |
| Excel Requirements and Mapping File | 79 |
| Requirement Views | 80 |
| Requirements Tree | 80 |
| Requirements Toolbar Functions | 82 |
| Requirements Document View | 83 |
| Requirement History | 83 |
| Tests | 85 |
| Configuring Test Settings and Integrations | 85 |
| Managing Products, Components, Versions, and Builds | 85 |
| Source Control Profiles | 93 |
| Step Properties | 108 |
| Attributes | 109 |
| Data Sources for Data-Driven Tests | 113 |
| Filters | 117 |
| Change Notification | 121 |
| Structuring Your Test Assets | 123 |
| Test Containers | 123 |
| Test Folders | 125 |
| Test Packages | 126 |
| Managing Tests | 129 |
| Test Types | 129 |
| Shared Tests | 172 |
| Editing Tests | 174 |
| Test Export, Update, and Import | 174 |
| Printing Tests | 178 |
| Test Runs Page | 178 |
| Test Issues Page | 179 |
| Versions | 180 |
| Configuring Test Properties | 182 |
| Test Properties | 182 |
| Test Attributes | 184 |
| Test Parameters | 185 |
| Test Attachments | 189 |
| Data-Driven Tests | 190 |
| Success Conditions | 193 |
| Executing a Trial Run of a Test | 194 |
| Libraries | 195 |
| Creating a Library | 195 |
| Managing Keyword Libraries and Library Properties | 195 |
| Making Libraries Accessible to Projects | 196 |
| Managing Keywords in a Library | 196 |
| Building a Keyword-Driven Library Using Java | 200 |

| | |
|---|------------|
| Keyboard Functions (Libraries) | 200 |
| Managing Shared Objects | 201 |
| Versions | 206 |
| Associating Requirements with Tests | 208 |
| Assigning Requirements to Tests | 208 |
| Sorting Assigned Requirements | 209 |
| Locating Assigned Requirements | 209 |
| Removing Requirement Assignments | 209 |
| Test Assigned Requirements Page | 210 |
| Test Reports | 210 |
| Status Reports | 210 |
| Progress Reports | 211 |
| Manual Test Reports | 211 |
| Baseline Tests Comparison Report | 212 |
| Test Run Comparison Report | 212 |
| Assigned Execution Plans | 212 |
| Viewing Assigned Executions | 212 |
| Test Views | 212 |
| Working with the Tests Tree | 213 |
| Test Toolbar Functions | 219 |
| Tests Document View | 221 |
| Working With Tests in Grid View | 221 |
| Execution Planning | 226 |
| Manual Execution Planning | 226 |
| Walkthrough | 226 |
| Test Selection | 228 |
| Test Assignment | 236 |
| Manual Testing | 244 |
| Configuration Testing | 256 |
| Creating a Configuration Suite | 257 |
| Creating a Configuration Suite from an Execution Plan | 257 |
| Adding Parameters to a Configuration | 257 |
| Removing Parameters from a Configuration | 258 |
| Assigning Keywords to a Configuration | 258 |
| Removing Keywords from a Configuration | 259 |
| Adding Manual Testers to Configurations | 259 |
| Removing a Tester Assignment from a Configuration | 259 |
| Assigning Mobile Devices to a Configuration | 260 |
| Configurations Suite Configurations Page | 260 |
| Configurations Page | 261 |
| Managing Execution Plans, Folders, and Configuration Suites | 262 |
| Creating Execution Plans | 262 |
| Creating an Execution Plan in Grid View | 263 |
| Test Status Calculation | 263 |
| Execution Plan Attributes | 263 |
| Execution Plan Parameters | 264 |
| Setup and Cleanup Tests | 266 |
| Configuring Deployment Environments | 267 |
| Configuring Execution Dependencies | 285 |
| Assigned Tests | 287 |
| Schedules | 291 |
| Executing Tests Assigned to Execution Plans | 295 |
| Managing Executions in External Execution Environments | 297 |
| Filters | 298 |
| Screen Capturing | 302 |
| Video Capturing | 302 |

| | |
|---|------------|
| Automatically Deleting Results and Result Files | 303 |
| Execution Planning Views | 304 |
| Working with the Execution Plans Tree | 304 |
| Execution Planning Toolbar Functions | 307 |
| Execution Plan Properties Page | 308 |
| Execution Planning Document View | 309 |
| Current Run Page | 310 |
| Execution Planning History Page | 311 |
| Working with Silk Performer Projects | 311 |
| Downloading Silk Performer Projects | 312 |
| Opening Silk Performer Projects | 312 |
| Accessing Execution Plan Run Properties | 313 |
| Executing Attended Silk Performer Tests | 313 |
| Editing Silk Performer Test Properties | 314 |
| Analyzing Silk Performer Test Results | 314 |
| Downloading Silk Performer Test Result Packages | 315 |
| Uploading Silk Performer Test Results | 315 |
| Working with Silk Performance Explorer | 316 |
| Silk Test Classic Tests | 316 |
| Adding a Silk Test Classic AUT Host | 316 |
| Automated Execution of Silk Test Classic Tests | 317 |
| Automated Execution of Data-Driven Silk Test Classic Test Cases | 317 |
| Specifying Agent Under Test (AUT) | 317 |
| Silk Test Classic Time-out Settings | 317 |
| Silk Test Classic Logs | 317 |
| Issue Tracking and Results Analysis | 319 |
| Analyzing Test Runs | 319 |
| Analyzing Manual Test Results | 319 |
| Analyzing Automated Test Results | 319 |
| Comparing Test Results of Multiple Test Runs | 319 |
| Changing the Status and Build of a Test Run | 320 |
| Viewing Test Run Details | 320 |
| Deleting Test Run Results | 320 |
| Deleting Runs and Result Files of Execution Plans | 321 |
| Execution Plan Run Results | 321 |
| Test Run Results | 322 |
| Viewing Execution Activities for Data-Driven Tests | 324 |
| Execution Runs Page | 324 |
| Tracking | 327 |
| Activities | 327 |
| Project Overview Report | 335 |
| Silk Central, Risk-Based Testing, and Quality Goals | 335 |
| Issues | 339 |
| Creating New Issues | 339 |
| Viewing Issue Statistics in Document View | 340 |
| Issues Document View | 340 |
| Viewing Issue Statistics in Details View | 340 |
| Issues Page | 340 |
| Assigning Existing Issues | 341 |
| Updating Issue States | 341 |
| Deleting Issues (Issue References) | 341 |
| Issue Manager | 341 |
| Reports | 398 |
| The Reporting Data Mart | 398 |
| Reporting Area | 418 |
| Code Analysis | 435 |

| | |
|---|------------|
| Default Reports | 445 |
| Administration | 458 |
| User Roles and Permissions | 458 |
| User Roles | 458 |
| Permission Definitions | 461 |
| User Accounts and Groups | 467 |
| Maintaining User Accounts | 467 |
| Maintaining Groups | 470 |
| Execution Environment | 473 |
| Managing Locations | 474 |
| Setting Up Execution Servers | 476 |
| Cloud Integration | 481 |
| Managing Device Providers | 484 |
| Managing Report Templates | 487 |
| Managing Custom Report Templates with BIRT | 487 |
| Adapting Existing Report Templates | 489 |
| Editing Report Template Properties | 490 |
| Downloading Report Templates | 490 |
| Uploading Report Templates | 490 |
| Updating Report Sources | 491 |
| Deleting Report Templates | 491 |
| Report Templates Page | 492 |
| Audit Log | 492 |
| Accessing and Viewing the Audit Log | 493 |
| Audit Log Page | 493 |
| Server Log Files | 494 |
| Downloading Server Log Files | 494 |
| Analyzing Server Log Files | 494 |
| Deleting Server Log Files | 495 |
| Log File Management | 495 |
| Front-End Server Log Page | 496 |
| Application Server Log Page | 497 |
| Execution Server Log Page | 498 |
| Integrating Task Management Tools | 499 |
| Integrating VersionOne in Silk Central | 499 |
| Managing Products, Components, Versions, and Builds | 500 |
| Managing Builds | 500 |
| Managing Components | 502 |
| Managing Platforms | 503 |
| Managing Products | 505 |
| Managing Versions | 507 |
| Maintaining Value Lists | 509 |
| Adding Value Lists | 509 |
| Editing Value Lists | 509 |
| Editing Individual Values | 509 |
| Activating and Deactivating Value Lists | 510 |
| Activating and Deactivating Individual Values | 510 |
| Sorting Values within Value Lists | 510 |
| Deleting Value Lists | 510 |
| Deleting Individual Values | 511 |
| List of Values Configuration Page | 511 |
| Configuring Global Schedules | 511 |
| Adding Global Schedules | 512 |
| Editing Global Schedules | 512 |
| Activating and Deactivating Global Schedules | 513 |
| Deleting Global Schedules | 513 |

| | |
|--|-----|
| Scheduling Exclusions | 513 |
| Scheduling Definite Runs | 514 |
| Schedules Configuration Page | 514 |
| Time Zones | 514 |
| Managing Projects | 515 |
| Working with Projects | 515 |
| Build Information | 524 |
| Comparing a Project with a Baseline | 525 |
| Selecting Projects | 526 |
| Project List | 526 |
| Switching to a Recently-Accessed Project | 527 |
| Project Settings | 527 |
| Project Templates | 591 |
| Integrations | 591 |
| Code Analysis Tools Integration | 591 |
| Issue Tracking Profile Integrations | 592 |
| Requirements Management Tools Integration | 605 |
| Source Control Profile Integrations | 619 |
| Test Automation Tools Integration | 633 |
| Continuous Integration Server Integrations | 650 |
| Communicating with an External System Over SSL | 651 |

Silk Central Help

Silk Central is a powerful, all-inclusive, software test-management tool. Silk Central builds quality and productivity into your testing process, speeding the delivery of successful software projects while minimizing the risks of application failures. It consolidates all the critical software-testing phases within a single, scalable, Web-based testing system, enabling your local and distributed software-development teams to share experiences, resources, and critical information.

What's New in Silk Central 20.6

Silk Central 20.6 introduces significant enhancements and changes.

Administration

Controlling Axis services for enhanced security

To enhance security, the SOAP services based on Axis 1.4 can now be deactivated. You can still use these services, however, they will be desupported in a future version of Silk Central. To access the services, go to <https://<hostname>/services> and <https://<hostname>/Services1.0/services>.

Unlimited filter criteria

You can now add an unlimited amount of selection criteria to your filters.

Precise permission management

The user permissions for tests and execution plans have been separated, which now lets you manage your permissions more precisely.

Dashboard

Enhanced Manual Tests Assigned to Me panel

The panel **Manual Tests Assigned to Me** lets you now group the panel columns. Also, the panel now provides the two new columns **Cycle Start Date** and **Cycle End Date**.

Requirements

Global list custom properties

The custom property type `Global List` is now also provided for requirement custom properties. The values of `Global List` custom properties are backed by a list of values.

Linked fields

Custom properties based on global lists can be grouped, which can be used to define depending fields, where the available values of one list depend on the selection in the other list.

Import .xlsx files that contain images

When importing requirements from an .xlsx file, the images contained in the file are imported by either inserting them into the requirement description or by adding them as attachments. If the image is located in a description cell and the MIME type is supported, the image will be inserted to the requirement description. Otherwise, the image is added as attachment and the attachment name is appended to the description.

Tests

Test Asset Workflow

Silk Central 20.6 provides a number of improvements for the test asset workflow:

- Silk Central now offers a dedicated permission that allows users to approve tests.
- You can now declare a test `obsolete` from all workflow states. Previously this was only possible from the workflow state `approved`.
- The workflow state of a test is now visible in the **Assigned Tests** grid of an execution plan.
- If not approved tests are assigned, a warning now displays on the **Run** dialog and when clicking **Start Testing Cycle**.
- When a test is started or synchronized, a message is added to the result documenting the workflow state.

Linked fields

Attributes based on global lists can be grouped, which can be used to define depending fields, where the available values of one list depend on the selection in the other list.

User list attributes

Silk Central 20.6 provides the new attribute type `User List`. The values of `User List` attributes are defined from system values according to the users assigned to a project.

Tests grid enhancements

The tests grid now displays information about the used source control profile and it allows you to filter for it.

Multi-edit in the tests tree

Silk Central now allows you to edit multiple objects in the tests tree through the context-menu. You can edit the property values and attributes values for several tests at once.

Execution Planning

Required attributes

Execution planning attributes can now be declared as required. When creating an execution plan, testing cycle, or a configuration suite, setting a value for these attributes is mandatory.

Download document view as .xlsx file

The content of the document view in the **Execution Planning** unit can now be downloaded as .xlsx file.

Linked fields

Attributes based on global lists can be grouped, which can be used to define depending fields, where the available values of one list depend on the selection in the other list.

User list attributes

Silk Central 20.6 provides the new attribute type `User List`. The values of `User List` attributes are defined from system values according to the users assigned to a project.

Issue Manager

Full user name for new inboxes

Silk Central now allows you to use the full user name when generating inboxes in Issue Manager.

Manual Testing

Alternative for JNLP

The functions **capture screen**, **record video**, and **code analysis**, provided in the **Manual Testing** window, are based on JNLP. For users who you cannot download the JNLP file (for example due to security restrictions), Silk Central now provides an alternative way to still use these features.

Executions

Version and build in Current Executions grid

In the **Current Executions** grid on the **Activities** page, now also the information about version and build is available for automated tests.

Deployment time of execution plans

The deployment time of execution plans, that is when the execution plan is first distributed to the execution server, is now also captured.

Change build information of finished executions

Silk Central 20.6 allows you to change the build information of already finished manual test runs. Doing so, however, requires a special permission and such action is specifically documented by adding information on who changed the version and build, when the values were changed, what the previous values were, as well as a mandatory comment by the user.

Integrations

Micro Focus Connect

Silk Central 20.6 further enhances the integration with Micro Focus Connect.

- It is no longer necessary to manually create reports, which simplifies setting up the integration.
- The relation between `test plan`, `test plan run` and `test run` is now added.
- The relation between `tests` and `issues` is now exposed.
- The optimized requirements synchronization enhances performance and reliability.
- The issue information from Issue Manager is now available to Micro Focus Connect.

UFT Developer

Silk Central 20.6 enhances support for UFT Developer through the test types `JUnit` and `MSTest`.

UFT Mobile

Silk Central 20.6 integrates with UFT Mobile (formerly Mobile Center).

JUnit 5.6

The JUnit plugin deployed with Silk Central 20.6 has been updated to support the result format of JUnit 5.6.

Atlassian Jira Software

Silk Central 20.6 provides a number of improvements for the requirements management integration and the issue tracking integration with Atlassian Jira Software.

- To authenticate Jira users, you can now use API tokens instead of plain passwords.
- When you create a new issue using the Silk Central UI, required fields are now marked and hence easy to spot.
- Support for Jira Software Cloud has been added.

Microsoft SQL Server Support

Silk Central now supports Microsoft SQL Server 2019.

Getting Started

Silk Central promotes product quality throughout the development cycle.

Today's e-business systems are increasingly complex, and reliability is more important than ever. Therefore, assuring product quality throughout the development cycle is an important key to success. The best way to check a product's quality over time is to perform key tests on a daily basis. Silk Central helps in this regard by automating test executions that follow freely configurable schedules, both during product development and after deployment. By providing reports with different levels of detail, checking the status of products in development is as straightforward as checking an HTML report in a Web browser.

With complex software projects, thorough testing of new builds is critically important. Silk Central saves time and man hours by automating this process.

Installing and Licensing Silk Central

For information regarding the installation and licensing of Silk Central, refer to the Silk Central Installation Help, which is also available from both the Silk Central installation DVD and the Silk Central download site.

Silk Central Concepts

This section explains the basic concepts of Silk Central, including the basic workflow, the available assets, and what they are used for.

Workflow

Silk Central's test management workflow consists of four major activities: Defining requirements, designing tests, execution planning and executing tests, tracking and analyzing test results.



Defining requirements



Designing tests



Execution planning and executing tests



Tracking and analyzing test results

Requirements



A requirement is a condition or capability that a user needs to achieve a specific task, which must be met by the application under test (AUT). Associate your requirements and tests to enable measuring how well your software covers the specified requirements. Multiple tests can cover a single requirement and multiple requirements can be covered by a single test. For additional information, see [page 36](#).

Tests



A test, or test case, is a set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement [after IEEE 610]. Silk Central supports various test types, and tests can be structured in several ways. For additional information, see [page 85](#).

Test types



Manual tests are step-by-step instructions for testers. Silk Central provides a UI to efficiently execute manual tests, including the option to record videos and capture screen images during test execution. If desired, manual tests can later be converted to automated tests. For additional information, see [page 129](#).



A *keyword-driven test* is an executable collection of keywords. A keyword-driven test can be played back just like any other test. Silk Central and Silk Test support the keyword-driven testing methodology and allow a very close collaboration between automation engineers and business analysts by having automation engineers develop a maintainable automation framework consisting of shared assets in the form of keywords in Silk Test. These keywords can then be used by business analysts either in Silk Test to create new keyword-driven tests or in Silk Central to convert their existing manual test assets to automated tests or to create new keyword-driven tests. For additional information, see [page 135](#).



Unified Functional Testing (UFT) software, formerly known as HP QuickTest Professional (QTP), provides functional and regression test automation for software applications and environments. For additional information, see [page 155](#).



JUnit is a unit testing framework for the Java programming language. Use this test type for JUnit, Silk4J, Selenium, and UFT Developer tests. For additional information, see [page 156](#).



NUnit is a unit testing framework for the Microsoft .NET programming language. For additional information, see [page 159](#).



MSTest is a command line utility from Microsoft that executes unit tests created in Visual Studio. Use this test type for MSTest, Silk4NET, and UFT Developer tests. For additional information, see [page 158](#).



The ProcessExecutor test type can be used to execute any command line. For additional information, see [page 160](#).



The Silk Performer .NET Explorer test type is used to execute .NET Explorer scripts or a specific test case of a script. For additional information, see [page 161](#).



Silk Performer projects can be integrated into Silk Central tests and directly executed through Silk Central. This allows for powerful test-result analysis and reporting. It also enables unattended testing, which means tests are run automatically by Silk Central based on pre-configured schedules. For additional information, see [page 162](#).



Silk Test Workbench is an automated testing tool that accelerates the functional testing of complex applications. Silk Test Workbench provides support for testing applications developed in a wide variety of development tools including Java, .NET, browser-based web applications, and COM components, including both ActiveX controls and automation objects. For additional information, see [page 164](#).

-  Silk Central's Silk Test Classic interface offers a reliable means of automating Silk Test Classic users. Each test case of a Silk Test Classic script executes within its own test execution and produces its own results. For additional information, see [page 162](#).
-  The Silk Test Classic Plan test type is used for Silk Test Classic test plans. A test plan usually is a hierarchically-structured document that describes the test requirements and contains the statements, 4Test scripts, and test cases that implement the test requirements. For additional information, see [page 162](#).
-  TestPartner is a Micro Focus product that tests Web- and Microsoft Windows-based applications that use Microsoft technologies. For additional information, see [page 166](#).

Test structure

-  Test containers are used to structure your test assets at the top level. Silk Central associates a defined source control profile and a product per container, which impacts what you can do with tests within a specific container. For additional information, see [page 123](#).
-  Use folders to structure your tests within a container. For additional information, see [page 125](#).
-  Test packages provide support for the structure of third-party test types in Silk Central, and consist of a package root as well as an arbitrary hierarchy of suite nodes and test nodes. For additional information, see [page 126](#).

Execution Planning

The **Execution Planning** area enables you to maintain control over test executions during development and testing. This includes activities like configuring execution plans, scheduling execution plans, assigning tests to execution plans, setting up execution plan dependencies, configuring execution-server deployment, and configuring dynamic hardware-provisioning with keywords.

-  An execution plan is a collection of assigned tests that are stored in a single test container. An execution plan can be run at configurable schedules and deployed on specified execution servers. For additional information, see [page 262](#).
-  Configuration testing is the process of testing a system with each of the supported software and hardware configurations. You can create configuration suites with a set of assigned tests, and all execution plans that you add to the configuration suite will also have the set of tests assigned. For additional information, see [page 256](#).
-  A testing cycle is a defined period in time consisting of a start date, an end date, and a list of manual testers. Create testing cycles to plan and execute specific sets of manual tests. For additional information, see [page 230](#).
-  Use folders to structure your execution plans, configuration suites, and testing cycles. You can also nest folders within folders. For additional information, see [page 262](#).

User Interface

This section provides an overview of the Silk Central user interface.

Tour of the UI

This topic provides an overview of the Silk Central user interface.

Basic UI Structure

The UI of Silk Central includes:

- The **Menu**: Placed across the top of the user interface. It gives you quick access to the functional units of Silk Central.
- The **Workspace**: In this example, the **Workspace** is the tabbed region showing the functional work area of the currently selected Silk Central unit. This view changes based on the unit you are working in.

The Units of Silk Central

The **Menu** gives you quick access to the functional units of Silk Central. It includes the following major sections:

Home

Displays the dashboard, which is your personal starting area. You can customize the dashboard by adding and arranging various panels, which give you the information you need for your daily work.

Requirements

Displays and enables you to maintain control over your project's requirements during development. You can perform the following tasks:

- Create, modify, and delete requirements.
- Associate tests with requirements.
- Track the change history of requirements.
- Generate test plans from requirement lists.

You can additionally access and manage the following:

- Properties of the requirements.
- Integrated external requirement-management tools.
- Requirement filters.
- Notifications that inform you about changes to the requirements.

Tests

Enables you to create and manage automated and manual tests. You can additionally access and manage the following:

- Libraries, out of which commonly-used manual tests, test steps, and keywords can be reused.
- Properties of the individual test steps for manual tests.
- Properties of keywords for keyword-driven tests.
- Test attributes.
- The data sources that can be used for tests.
- Source control profiles, in which these tests are stored.
- Test filters.
- Notifications that inform you about changes to the tests.
- Version and build of the product that is tested with a specific test.

Execution Planning

Displays and allows you to manage the executions that execute the tests. You can perform the following tasks:

- Configure execution plans.
- Assign tests to execution plans.
- Set-up execution plan dependencies.
- Configure execution-server deployment, including the execution servers on which the tests are executed.
- Plan and manage the manual execution of tests.

You can additionally access and manage the following:

- Execution filters.
- Execution schedules.

- Version and build of the product that is tested with a specific execution of a test.
- Notifications that inform you about changes concerning the execution process.

Tracking

Displays the **Activities** page, the **Cross-Project Activities** page, the **Project Overview Report**, and **Quality Goals**.

The **Activities** page displays the activity for recently-executed, current, and upcoming execution plans on a per-project basis.

Issues

Displays and enables you to manage the issues that are related to the active project.

You can additionally access and manage the following:

- Projects in Issue Manager, when issues reside in Issue Manager.
- External issue tracking profiles, in which issues may possibly reside.

Reports

Displays and enables you to manage the reports which provide you with information on the requirements, tests, and execution plans. You can perform the following tasks:

- Generate reports.
- Download report templates.
- Edit report parameters.
- Create new reports based on pre-installed templates.

Additionally provides access to code analysis functionality information, with which you can evaluate the degree to which the code in your Application Under Test (AUT) is covered by test cases. You can then make informed estimates regarding effort, cost, and risk associated with specific code changes.

Global Search

The **Search** field allows you to perform a search across several assets in Silk Central. The following assets are considered when submitting a global search:

- Requirements and their properties
- Tests, folders, test containers and their attribute values, properties, and parameters
- Test steps and their properties
- Execution plans, configuration suites, testing cycles, folders and their parameters
- Library folders, steps, and shared steps
- Issues (searched by synopsis and description)



Note: The search index is updated in the background on a regular basis. The search results may occasionally be slightly outdated.

Search rules:

- The search is case-insensitive.
- All entered words are searched for separately. Items match the search if at least one word is found, and the more words match the query, the higher the rating of the match is.
- Prefix words that must match the query with a +. For example "+test +type" will only return results that match both "test" and "type".
- Prefix words that may not match the query with a -. For example "test -copy" will return results that match "test", but not "copy of test".

- To search for an exact match, put your search words inside quotes, for example "customer account management".
- To limit the search to the item's name, or to the description, prefix the search word with `name:` or `description:`, respectively. For example `name:test` will only return results where the item's name contains the word `test`.
- Common English words like 'a', 'the', 'and' are ignored.

Example:

```
name:"test item" -description:closed
```

This will return items where the name contains the phrase "test item" and the description does not contain the word "closed".

Projects

Displays and enables you to manage all projects in your Silk Central installation from a high-level test-manager's view. Additionally enables you to switch between projects, and manage project settings.

User

Displays and enables you to manage user settings. Additionally enables the following:

- Log out of Silk Central.
- Print the current Silk Central page.
- Bookmark the current Silk Central page. This is especially useful for bookmarking reports, where the current parameters are saved in the book-marked URL.

Administration

Displays and enables you to manage the administration settings. For additional information, see the *Administration* topics in this Help.

Help

Click to view context-sensitive help for the current page. Additionally provides access to the following:

- The Silk Central documentation.
- Tools that are shipped with Silk Central.
- About. Shows version and license information.

Context Menu Commands

Silk Central supports Windows-style context menus across many test management elements, for example tests, requirements, execution plans, folders, containers, reports, and more. Available through right mouse-click, context menu commands typically include those commands that are available from each unit's toolbar. For elements listed in tree views, context menus offer commands for expanding and collapsing tree view elements. Commands that are not available to selected elements are disabled.

Dashboard

The dashboard in Silk Central is your personal starting area. You can customize it by adding and arranging various panels, which give you the information you need for your daily work.

When you log in to Silk Central for the first time, the dashboard shows just the **Introduction** panel. On the top, your last login date and time is displayed. You can quickly **Resume where you left off** by clicking this link.

Click the buttons on the top, to perform the following actions:

- Click **Add Panel** to add further panels to the dashboard.
- Click **Change Layout** to change the layout of the dashboard.
- Click **Reset Dashboard** to set the dashboard to the default view. All the panels you added and your customized settings are removed.

To arrange your panels, click on the header of a panel and drag it to the desired position. Resize your panels by dragging a panel's footer (only panels with lists are resizable).

Click the buttons in the header of a panel to perform the following actions:

- Click  to remove the panel from the dashboard.
- Click  to configure the settings of the panel.
- Click  to save the content of the panel as a PDF.
- Click  to print the content of the panel.

You need certain permissions to view or edit the content of the different panels.

Adding Panels to the Dashboard

To add a panel to the dashboard:

1. In the menu, click **Home > My Dashboard**.
2. Click **Add Panel** on the top left. The **Add Panel** dialog box appears, listing the available panels along with a short description for each panel.
3. Select a panel.
4. Click **OK**.
5. If required, select a **Project** and a **Time Span** for the panel. Select **<Use current project>** for the panel to always display the data of the currently active project.
6. Click **OK**.

For most panels, you need to configure a project, as the panel will only show data for a certain project. You can add several panels to show appropriate data for other projects side-by-side, or configure **<Use current project>** for the panel to always display the data of the currently active project. Click  (**Configure**) in the header of a panel to configure a different project. You need certain permissions to view or edit the content of the different panels.

Dashboard Panels

For most panels, you need to configure a project, as the panel will only show data for a certain project. You can add several panels to show appropriate data for other projects side-by-side, or configure **<Use current project>** for the panel to always display the data of the currently active project. Click  (**Configure**) in the header of a panel to configure a different project. You need certain permissions to view or edit the content of the different panels.

You can add the following panels to your dashboard:

| Panel | Description |
|---------------------------|--|
| Assigned Issues | This panel shows all issues (that are assigned to a test) of the selected project. If you click the external ID, the issue opens in the external issue tracking system. |
| Custom Information | This panel shows customizable information about a certain project. Users with the Manage projects permission can add content like text, images or hyperlinks to the panel. Click Edit Content to open the HTML editor. Users without the Manage projects permission will just see the content. The panel can be used to show news, descriptions or any other kind of information relevant to the project. |

| Panel | Description |
|--|---|
| Introduction | This panel gives you a quick introduction to Silk Central. Click the links under What's New? to get additional information about the latest features of Silk Central. You can also join the Micro Focus Community or contact the Micro Focus SupportLine . |
| Issue Life Cycle | This panel shows the number of Open, Fixed, Verified, Closed and Deferred issues for a defined issue tracking profile. Select a Product from the list. To define an issue tracking profile, click Issues in the menu, click Issue Tracking Integrations and click New Profile . The panel will only show data, if there is a profile defined. |
| Issues Created per Tester | This panel shows the number of issues created by testers as a bar graph. When printed or downloaded as a PDF, the numbers will be shown in a table as well. |
| Manual Tests Assigned to Me | <p>This panel shows all tests of the selected project that were assigned to you.</p> <p>If the manual test execution is organized with the help of testing cycles, the list also includes all tests that are assigned to No specific tester on the Manual Execution Planning page. However, this is only true for testing cycles you are assigned to as a tester. It is a typical teamwork scenario that testers decide on their own which test to execute.</p> <p>If the manual test execution is organized with the help of execution plans, the list also includes the tests from all execution plans that have no tester assigned.</p> <p>Click  (Continue Manual Test) to open the Manual Testing window.</p> <p>Click  (View Manual Test Results) to open the view and see all data and attachments about the test.</p> <p>Click  to get detailed information about the Test, the Execution Plan or the Execution Plan Parent.</p> <p>Right click a test and click  to download a manual test (in Excel format) to execute it offline (multiple tests can be selected with Ctrl+Click or Shift+Click). After you execute the test, click the Upload Offline Manual Test Results button to upload the results.</p> <p>If you start a testing cycle (on the Manual Execution Planning page) before its start date, the tests of this testing cycle already display in the panel. You can make those tests invisible in the panel until the start date is reached: Click  in the header of the panel and check Hide future testing cycles.</p> |
| Offline Manual Testing Overview | This panel shows the tests that are downloaded by testers and allows the Test Manager to abort offline executions to free the license. |
| Planned vs. Actual Execution Time | This panel shows if the testers of a testing cycle are on schedule with the test execution. If the actually used time plus the remaining planned time is greater than the capacity, the testing cycle is on risk to be not finished on time. |
| Quality Goal Progress | This panel shows a graph with two bars for each value of the selected quality goal. The upper bar shows the number of tests that must be executed to meet the quality goal. The lower bar shows the number of the Passed, Failed, Not Executed , and N/A (not available) tests with that quality goal value. If the lower bar is shorter than the upper bar, too few tests are assigned to execution plans. This means that the quality goal cannot be reached. Select a Quality Goal from the list. |
| Report Subscription | This panel displays the results of a report subscription. The report subscription must already be defined and you need to select Provide for dashboard and external access from the Edit Subscription dialog box to make it available. |
| Requirements Coverage Status | This panel shows the requirement coverage of the selected project, broken down into Passed, Failed, Not Executed , and Not Covered requirements. Click  to go to the Requirements Document View . |
| Silk Central Connect | Silk Central Connect combines aspects of test execution management and configuration testing into an easy to use interface, providing the following advantages: |

| Panel | Description |
|---|--|
| | <ul style="list-style-type: none"> • Simple execution of all your automated unit tests on a variety of <i>configurations</i>, which are combinations of operating systems and Web browsers. • Leverages the advantages of the Amazon Web Services, enabling you to easily access a variety of configurations without any upfront investment. • Tight integration between Silk Central Connect and Silk4J for easy test creation, maintenance, and execution. • Side-by-side result analysis, enabling you to compare all of your tests across the different configurations. <p>For detailed information, see Silk Central Connect.</p> |
| Silk Performer Performance Trends | The Silk Performer Performance Trends panel displays the current status of the selected Silk Performer load test and its available results, including trend information over the last runs. |
| Status of Currently Running Manual Tests | This panel shows the status of all currently running manual tests in the specified project. Test managers are able to see which tests are currently blocked and they can monitor the status of already executed tests in running executions. |
| Testbook | This panel shows you real-time updates about all activities that take place during manual testing. Activities that are logged in the Testbook are starting and finishing tests and editing running testing cycles. For more information, see <i>Testbook</i> . |
| Testing Cycle Progress | This panel shows a burn-up chart of Passed , Failed , In Progress , and Not Executed tests for the defined testing cycle. It also shows the milestones (as diamonds), if you have specified any. Select a Testing Cycle from the list. |
| Testing Cycle Result Summary | This panel shows the status of the selected testing cycle and the progress status of the individual testers that are assigned to the testing cycle. The bars show in the following order: the number of Tests failed , Tests remaining , and Tests passed . The tester with the most failed tests displays on the top of the list. The numbers to the right of the bars show the Tests completed (left number) and the Tests assigned (right number). No specific tester is treated and ordered like all other testers. Click  to go to the Test Assignment view of the Manual Execution Planning page. |
| Testing Cycle Timeline | This panel shows all testing cycles of the selected project as bars on a timeline. When you click on a bar, you are directed to the Manual Execution Planning page. The bars have different colors, depending on the current status of the testing cycle: blue (in progress), orange (not yet started), gray (finished). Move your mouse over the bars to get detailed information about the testing cycle. |
| Testing Progress Across Testing Cycles | This panel shows the time in hours for the Planned tests (visualized by a line) and for the In Progress and the Completed tests (both visualized by stacked areas). The panel helps a test manager to predict if the testing team will be able to finish the tests in time. The start date of the chart is the start date of the first testing cycle. The end date of the chart is the end date of the last testing cycle. The panel does not show how long it actually took the testing team to execute the tests. |
| Volatile Tests | This panel shows how often tests changed their status within an execution plan. Tests without status changes and test packages do not display in the panel. Tests display according to the selected project and time span. The panel makes test managers aware of such volatile tests, which indicate either an unstable test or a tested area where bugs occur frequently. |

Dashboard Panel Permissions

To view or edit the content of a certain dashboard panel, you need the following permissions:

| Panel | Required permission |
|---------------------------|--|
| Assigned Issues | View tests. |
| Custom Information | Manage projects required for editing. |

| Panel | Required permission |
|---|--|
| | No permission required for viewing. |
| Introduction | No permission required. |
| Issue Life Cycle | View project settings. |
| Issues Created per Tester | View tests or view executions. |
| Manual Tests Assigned to Me | Execute tests. |
| Offline Manual Testing Overview | View tests or view executions. |
| Planned vs. Actual Execution Time | View testing cycles and configurations. |
| Quality Goal Progress | View quality goals. |
| Report Subscription | No permission required. |
| Requirements Coverage Status | View requirements. |
| Silk Performer Performance Trends | View tests or view executions. |
| Status of Currently Running Manual Tests | View tests or view executions. |
| Testbook | No permission required. |
| Testing Cycle Progress | View testing cycles and configurations. |
| Testing Cycle Result Summary | View testing cycles and configurations. |
| Testing Cycle Timeline | View testing cycles and configurations. |
| Testing Progress Across Testing Cycles | View testing cycles and configurations. |
| Volatile Tests | View tests or view executions. |

Adding a Report to a Dashboard Panel

Using any report available in the **Reports** unit, the **Report Subscription** panel of your dashboard allows you to display an existing report. First, create and configure your report, and then add the panel to your dashboard.

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Subscriptions** tab.
4. Click **Add Subscription**. The **Edit Subscription** dialog box opens.
5. Give your subscription a name in the **Name** field.
6. To modify the subscription parameters, click **Edit Parameters**. The **Edit Parameter** dialog box opens. Change the **Value** and click **OK**.
7. Select a schedule from the **Schedule** list. If the schedule you want does not exist, you need to create it. You can also select no schedule, for example for subscriptions that will only be triggered based on the outcome of a test execution.
8. Click **Provide for dashboard and external access** to make the report available in a dashboard panel.
9. Click **OK**. The new subscription is listed in the **Subscriptions** list.
10. In the menu, click **Home > My Dashboard**.
11. Click **Add Panel** on the top left. The **Add Panel** dialog box appears, listing the available panels along with a short description for each panel.

12. Select **Report Subscription**. The **Configure Panel** dialog box appears, listing all available report subscriptions.

13. Select a report subscription from the list and click **OK**. The panel is added to your **Dashboard**.

Browser Settings

To optimize the functionality and performance of Silk Central, adjust the following settings in your browser:

- Make sure that you enabled the cache of your browser.
- If you have problems accessing Silk Central through the browser, add the Silk Central server to the list of trusted sites in your browser.

For a list of supported browsers, refer to the Release Notes. In the menu, click **Help > Documentation**. Click **Silk Central 20.6 Release Notes**.

Login and Logout

The procedures in this section explain how to log in to and out of Silk Central.

Logging in to Silk Central

To log in to Silk Central:

1. Navigate to the URL of your Silk Central installation.
2. On the Silk Central login page, enter your user name in the form `<client>\<username>` and your password.

When you login to Silk Central with a user of the default client, you do not need to enter the client name. Just enter your user name.

Log in as Super User with the following default credentials:

- username: `admin`
- password: `admin`

3. If you check the **Remember login** check box, you will stay logged in until you close the browser tab. The next time you access Silk Central, you will automatically be logged in again.
4. Click **Login**.

You will be directed to your personal **Dashboard**. You can quickly **Continue working in your last visited area** by clicking this link on the top of the **Dashboard** page.

Logging out from Silk Central

To log out from Silk Central:

1. In the menu, go to **User:<Username>**.
2. Click **Log out**.

Accessibility

Silk Central offers accessibility for users with a visual disability.

Silk Central users can use the keyboard to navigate to any control or asset in any unit of the Silk Central UI. Additionally, users can use a screen reader to identify a control or asset in the UI and to understand what the control or asset is intended to do.

Quick Start Tasks

Quick start tasks are high-level overviews of the main tasks that you will likely need to perform with Silk Central. These tasks can serve as tutorials in guiding you step-by-step through the best practice usage of the core functionality of Silk Central. For in-depth information, see the related sections in this help: [Requirements](#), [Tests](#), [Execution Planning](#), and [Issue Tracking and Results Analysis](#).

Configuring Required Assets - Quick Start Task

Before you can start building up your testing framework, you need to at least configure a [project](#) and a [product](#) and, if you plan to create automated tests, also a [source control profile](#). You may also need to configure certain other settings, like filters, properties, attributes, etc. You need SuperUser, administrator, or project manager privileges to perform these tasks.

To configure the required assets, you will need to perform the following tasks.

Adding Projects

Silk Central manages requirements, tests, their execution and the results in the context of a project. Therefore the first thing you have to do is to create a project.

To create a project:

1. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
2. Click **New Project**. The **Project Settings** page displays.
3. Type a **Project name** and **Description**.
4. Select the **Project Owner**.
5. *Optional:* Enter **Tags** for your project. If you have many projects, project tags can help you filter them, using the **Tags** column on the **Project List** page.
6. *Optional:* To create a project based on the Agile project template, choose *Agile Project Template* from the **Project Template** list box.
7. To initialize the project as an Issue Manager project, click **Initialize new project** or **Reuse existing project**. For more information, see *Initializing Projects as Issue Manager Projects*.
8. In the **Shared Tests** section, define whether the project is sharing tests or not. Select one of the following options:
 - **No test sharing:** This project neither shares tests with other projects nor consumes tests from other projects.
 - **Sharing tests:** This project is sharing tests that other projects can consume.
 - **Consuming shared tests from:** Select a project from which the edited project consumes shared tests.
9. The **Groups** section includes lists of available and assigned groups. Assign the groups as required. Use **Ctrl+Click** or **Shift+Click** to select multiple groups. You can also import groups from an LDAP server by clicking **Import LDAP Group**. For detailed information, see *Importing Groups from LDAP Servers*. If no user groups exist, you may assign them later after you have created them. You can also configure the group/project assignment on the **Group Settings** page. Privileges vary based on user roles. For information about user privileges, see *User Roles and Permissions*.
10. The **Locations** section includes lists of available and assigned locations. Assign the locations as required. Use **Ctrl+Click** or **Shift+Click** to select multiple locations. If no locations exist, you may assign them later after you have created them. You can also configure the location/project assignment on the **Location Settings** page. For detailed information, see *Managing Locations*.
11. Click **Save** to save your settings. You are returned to the **Project List** page where the new project is listed.

Adding Products

Silk Central tracks results of test executions in the context of versions and builds of one specific product. Therefore a **Product** needs to be defined before tests can be created.

To add a new product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click **New Product**. The **New Product** dialog box displays.
4. Type a name for the new product in the **Name** field.
5. Type a description for the product in the **Description** field.
6. Check the **Active** check box to make this product available for association with tests.
7. Click **OK** to save the product.



Note: When creating a product, Silk Central automatically creates a new default version 1.0 and a new default build 1 for the new product.

Configuring Source Control Profiles

For automated tests, another requirement for test creation is a **Source Control Profile**. Automated tests in Silk Central actually represent test scripts that are located in source control systems, which are accessed via such source control profiles.

For detailed information about source control profile configuration, see [Source Control Profiles](#).

Defining Requirements - Quick Start Task

Manage your requirements so you can associate them with your tests to enable measuring how well your software covers the specified requirements. Integrating external requirements management enables you to coordinate the requirements management features of Silk Central with other tools you may already be working with.

To manage your requirements, you will need to perform some or all of the following tasks.

Creating Requirements

Associate your requirements and tests to enable measuring how well your software covers the specified requirements.

To create a new requirement:

1. In the menu, click **Requirements > Details View** .
2. In the **Requirements** tree, select a requirement or the root node.
3. Click  (**New Requirement**) on the toolbar to create a new requirement on the same hierarchical level, or  (**New Child Requirement**) to create it hierarchically beneath the selected requirement. The **New Requirement** dialog box appears.
4. Type a **Name** and a **Description** for the requirement.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for description fields.

5. *Optional:* Uncheck the **Inherit from parent** check boxes. In this case the child requirement will not inherit the properties from the parent requirement. By default, all check boxes are checked.



Note: In the **Details View** on the **Properties** page and in the **Document View**, inherited properties are marked with asterisks (*).

6. Select the appropriate **Priority**, **Risk**, and **Reviewed** status from the lists.

7. If custom requirements were defined, they are listed below the **Reviewed** list. Type in the corresponding field any custom property data that you want to track with this requirement. This is disabled for inherited properties. For details regarding creating, editing, and deleting custom requirement properties, see *Custom Requirement Properties*.
8. To finish creating a new requirement:
 - Click **OK** to create the requirement and to close the dialog box.
 - Click **OK and New Requirement** to create the requirement and to keep the dialog box open to enter data for another requirement.
 - Click **OK and New Child Requirement** to create the requirement and to keep the dialog box open to enter data for a child requirement.

Generating Tests from Requirements

You can generate tests directly out of the **Requirements** tree and assign tests to specific requirements. The **Requirements** tree serves as a template for the test folder/test structure of the new **Tests** tree.

To generate a new test from the **Details View**:

1. In the menu, click **Requirements > Details View**.
2. Right-click the requirement or project node that you want to convert into a test and select **Generate Tests**. The **Generate Tests from Requirements** dialog box appears. This dialog box enables you to specify whether the leaves, which means the lowest-level nodes, of the selected requirements sub-tree should be converted into tests or test folders and whether the tree should be generated into a new test container or an existing container.
3. Enter a name for the new test container in the **Enter Name** field and select a product from the **Select Product** list to create the container within the active Silk Central project.

The **Select Product** list is populated with the products that are configured by a project manager. For detailed information, see *Managing Products* or ask your project manager.
4. If you have defined a source control profile, select the source control profile you want to use for managing the test sources from the **Select Source Control Profile** list.

For detailed information on source control profiles, see *Source Control Profiles* or ask your Silk Central administrator.
5. To include all child requirements of the selected requirement in the test, check the **Include Child Requirements** check box.

The check box is checked by default.
6. To have the new tests automatically assigned to the requirements from which they are created, check the **Assign newly generated tests to Requirements** check box.

If this option is not selected, tests must be manually associated with requirements.



Note: This option is not available when checking **Generate test folders from requirement tree leaves**.

7. Click **OK** to create the test.

The new test has the same structure as the **Requirements** tree.
8. A message box displays. Click **Yes** to view the test in the **Tests** area, or click **No** to remain in the **Requirements** area.

Integrating External Requirements Management Tools

Integrating external requirements management enables you to coordinate the requirements management features of Silk Central with other tools you may already be working with.

For detailed information about integrating external requirements management tools, see [Requirements Management Tools](#).

Designing Tests - Quick Start Task

Design your tests to cover all related features with useful tests.

To design your tests, you will need to perform some or all of the following tasks.

Structuring Your Test Assets

Silk Central tracks results of test executions in the context of versions and builds of one specific product. Therefore a **Product** needs to be defined before tests can be created. For automated tests, another requirement for test creation is a **Source Control Profile**. Automated tests in Silk Central actually represent test scripts that are located in source control systems, which are accessed via such source control profiles. Both product and source control profile information are specified on **Test Container** level, which therefore represents the top-hierarchy node of a test assets structure.

To get started, we'll simply create one container:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select the root node to which you want to add the new container as a sub-node.



Note: Containers cannot be nested, but you can add sub-folders to containers.

3. Click  on the toolbar or right-click on the node and choose **New Test Container**.

You can also right-click on the node and choose **New Test Container**.

A new container root node is added to the **Tests** tree and the **New Test Container** dialog box displays.

4. Type a **Name** and **Description** for the container.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. Select any predefined **Product** that is to be associated with this test container from the list box. For details on adding a product profile, see *Adding Products*.
6. Select a predefined source control profile from the **Source Control profile** list box.

If you do not intend to check out any test sources for the test container, for example if you only have manual tests, select ---.



Note: The Silk Central execution servers retrieve the program sources for the tests included in the container from the defined source control profile. Depending on the specified path in your source control profile, you can specify to use a sub-node in the **Root node** field of the test container.

7. *Optional:* Check the **Clear working folder before a test is executed** check box.

If checked, files that have been removed from the source control will also be removed from the working folder. Default is checked.

8. *Optional:* Click **Browse** to select the root node within the selected source control profile.

Test sources are within this root node. The root node is used on the execution servers to retrieve the sources of a test.

9. *Optional:* When Silk Central is integrated with Silk Performer, you can define the **Custom data directory** and **Custom include directory**.

In Silk Performer, the `Include` directory is divided into a `System Include` directory and a `Custom Include` directory. The `Data` directory is divided into a `System Data` directory and a `Custom Data` directory. For additional information, refer to the Silk Performer documentation.

10. The **Hidden Test Properties** portion of the dialog box allows you to specify the test property types that are to be displayed on the test container's **Properties** page, and the **Properties** pages of all test folders within the container.

These settings do not affect the display of individual tests.

11. Click **Edit** to adjust the hidden test property settings. The **Hidden Test Properties** dialog box displays.
12. Uncheck the check boxes of all test types for which you want to have properties displayed.
13. Click **OK** to save your settings and close the **Hidden Test Properties** dialog box.
14. Click **Save** to save your settings.

Creating Tests

Create a test to test a specific feature of your software.

Silk Central supports several test types out-of-the-box. See [Managing Tests](#) and select the test type that you want to create, then follow the instructions in the referenced topics.

Assigning Requirements to Tests

Associate your requirements and tests to enable measuring how well your software covers the specified requirements.

To manually assign requirements to tests:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select the test to which you want to assign requirements.
3. Click the **Assigned Requirements** tab.

 **Note:** If you have created a requirements filter, you can select it from the filter list above the **Requirements** tree. To create a new requirements filter, click **Requirements > Details View** in the menu and click  on the toolbar.

All requirements that are available for assignment are displayed in the **Requirements** tree.

4. Select the requirements that you want to assign.

You can use your keyboard's **Ctrl** and **Shift** keys to select multiple requirements using standard browser multi-select functions.

5. Click **Assign All Filtered** to assign all filtered requirements to the selected test, or drag your selection and drop it into the **Assigned Requirements** area, or click  to the left of any requirement. Assigning all filtered requirements is only enabled if a filter is selected and if you have either **Manage requirements** or **Manage assets** permission.

 **Note:** Newly generated tests can automatically be assigned to the requirements from which they are generated by checking the **Assign newly generated tests to requirements** check box on the **Generate Tests from Requirements** dialog box. This is the default behavior.

Execution Planning and Executing Tests - Quick Start Task

To plan and execute your tests, you will need to perform some or all of the following tasks.

Planning and Executing Automated Tests

Create an execution plan to execute related tests together.

Creating Execution Plans

Create an execution plan to execute related tests together.

1. In the menu, click **Execution Planning > Details View**.
2. Select a folder or the project node in the **Execution Plans** tree.
3. Click  (**New Child Execution Plan**) on the toolbar or right-click the folder, testing cycle, or node and click **New Child Execution Plan**. The **New Execution Plan** dialog box appears.

4. Type a name and description for the execution plan.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. Select a test container from the **Test Container** list.



Note: You can assign only one test container to an execution plan. Only tests from this test container can be executed.

The latest version and build that are defined within the product that the container belongs to are populated automatically in the **Version** and **Build** lists.

6. Select a product **Version** and **Build** from the lists.

These are used when a new run of this execution plan is started. Alternatively, you can check the **Read from Build Information File** check box. In this case the version and build are read from the build information file on the application server at the beginning of each run. If a build information file is available on the execution server, this file is used by default for the test run, overriding the settings on the **New Execution Plan** dialog box.

7. Select a **Priority** for the execution plan from the list.

For automated execution plans, when more than one execution plan is queued but only one execution server is available, the **Priority** determines which execution plan is executed first. The same is true if execution servers are available but multiple execution plans require an unavailable mobile device.

8. In the **Source Control Label** field you can optionally specify that the execution plan is of an earlier version than the latest version. The label must refer to a version in the source control system that the test container is associated with. If this field is left blank, the latest version of the execution plan will be fetched. The **Source Control Label** property is only enabled if the associated test container uses a source control profile that supports versioning. Make sure to have enough free disk space on the execution server or servers when working with multiple versions of source files. Each version will be saved in its own folder on every execution server.



Tip: If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan.

9. Specify all **Required attributes**, if there are any. Required attributes must be specified when creating a new execution plan. Required attributes that are not yet set display with an orange background and a red border.



Note: If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.



Note: If an attribute is set to **Inactive** (under **Execution Planning > Attributes**), it is not required when creating a new element.

10. Click **OK** to update the **Execution** tree with the newly created execution plan.

Manually Assigning Tests to Execution Plans

Manually assign a test to an execution plan to add it to the included tests.

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan to which you want to assign the selected tests.
3. Click the **Assigned Tests** tab.
4. Click the **Manual assignment** option. All tests of the test container which is associated with the selected execution are displayed in the **Tests** tree. If you have created a test filter, you can select it from the filter list above the **Tests** tree. To create a new test filter, click **Tests** in the menu and click  (**New Filter**) in the toolbar.
5. Select the tests that you want to assign.

You can use your keyboard's **Ctrl** and **Shift** keys to select multiple tests using standard browser multi-select functions.

6. Drag your selection and drop it into the **Assigned Tests** area, or click  left of any test.

Assigning a folder or the top-level container assigns all child tests of that parent to the selected execution plan.

 **Tip:** Right-click a node to expand and collapse it.

Starting Execution Plans

Execute individual execution plans to run them independent of a schedule.

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the project, execution plan, folder, or configuration suite that you want to execute, or use **Ctrl+Click** or **Shift+Click** to select multiple elements.
3. Click  (**Run**) on the toolbar. This is disabled for testing cycles, execution plans within testing cycles, and unassigned tests. The **Run** dialog box appears.
4. Define which tests you want to execute in the **Run** dialog box.
5. Check the **Go to Activities page** check box to advance to the **Activities** page after you define tests for execution.

Planning and Executing Manual Tests

This walkthrough shows you features of **Manual Execution Planning** using an example. The assumptions for the example include:

- The requirements are fully documented and categorized according to a `Risk` property.
- The manual tests are fully documented, assigned to requirements, and categorized according to an `Importance` attribute which defines the necessity for executing a specific test.
- Quality goals, which are also called *project exit criteria*, have been defined based on the requirements property `Risk` and the test attribute `Importance`. These quality goals specify the levels of testing required.
- The only tests to plan are the ones that best fulfill the quality goals.

Overview

The **Manual Execution Planning** page consists of four areas presented on two screens. On the first screen, the **Test Selection** screen, you can select and schedule your tests for a testing cycle. On the second screen, the **Test Assignment** screen, you can assign the tests of the selected testing cycle to the testers who will execute them.

The **Test Selection** screen is divided into three areas:

- The **Filter Criteria** area on the left side. With the filter capabilities, you can narrow down the tests shown in the next area.
- The **Matching Tests** area in the middle. This list shows all available tests. When you apply filters, just the filtered tests are shown.
- The **Testing Cycles** area on the right side. Here you can assign tests from the list in the middle to a certain testing cycle, which is a defined time period.

To access the **Test Assignment** screen, click **Test Assignment >>>** on the top right. This screen is divided into two areas:

- The **Testing Cycles** area on the left side. This is the same area as on the **Test Selection** screen, it is shown on both screens.

- The **Assigned Tests** area on the right side. This list shows the tests, which have been added to the currently selected testing cycle. Here you can drag the tests from the list to the **Testing Cycles** area and drop them on a tester.

Creating a Testing Cycle

The first step is to create a testing cycle. A testing cycle is a defined period in time consisting of a start date, an end date, and a list of manual testers. We will name our testing cycle `Week 21 - Team A`.

1. In the menu, click **Execution Planning > Manual Execution Planning**.
2. In the **Testing Cycles** area on the right side, click **New Testing Cycle**. The **New Testing Cycle** box appears.
3. Enter a name for the testing cycle, in our case `Week 21 - Team A`.
4. Optionally enter a description.
5. Click  next to **Start date** to set the start date of the testing cycle.
6. Click  next to **End date** to set the end date of the testing cycle.
7. In `hh:mm` format, enter the amount of time that you want to provide for this testing cycle (**Capacity**), in our case, we entered `35:00`.
8. Click **OK**.

Our testing cycle now appears in the **Testing Cycles** area. It shows that currently no testers are assigned. The next step is to assign testers.

Adding Manual Testers

The next step is to add manual testers to our testing cycle. We are also going to define the capacity each manual tester has for this testing cycle.

1. In the header of our testing cycle `Week 21 - Team A`, click **... > Edit Assigned Testers**. The **Edit Assigned Testers** dialog box appears.
2. Click **Add/Remove Testers**. The **Add/Remove Testers** dialog box appears.
3. Select a user group from the **Available** list.
4. Select one or more users from the user group in the **Available** list.
5. Click **Add** or **Add All** to add users to the **Selected** list.
6. Click **OK**. The users are now added to the list in the **Edit Assigned Testers** dialog box.
7. For each user that was added, provide a capacity for the testing cycle in the **Capacity** field using the `hh:mm` format.
8. Click **Close**.

Now our testing cycle shows the available resources, both for the whole testing cycle as well as for each manual tester. A burn-down chart displays the assigned tests that are not yet completed. After we have created our testing cycle and added manual testers, we can determine which tests need to be done in the next step.

Selecting Tests and Scheduling

This topic describes how to assign tests to our testing cycle. The goal is to select those tests that best cover the defined quality goals. One of the quality goals is based on the requirement risk. The typical practice is to test things with high risk first.

1. In the **Filter Criteria** area on the left side, under **Quality Goals**, check the **Requirement Risk** check box. The list of **Matching Tests** is narrowed down to the tests that are critical.
2. In the **Matching Tests** area, open a column header's context menu to sort, filter or group tests. That helps to select the tests that should be scheduled.

3. Select one or more tests, drag them to the **Testing Cycle** area on the right side and drop them to our testing cycle. To assign tests to a testing cycle, you can also right-click on the selected tests and select **Assign tests to cycle**. When adding tests to a testing cycle, the indicator for already used resources is updated. Move the mouse over this indicator for detailed information about the overall capacity, the planned time, and the time left.
4. After scheduling all tests with a critical risk, schedule those with a high risk. After that, schedule those with a medium risk, and so on.

The tests assigned to our testing cycle have moved to the **No specific tester** row. In the next step, we will assign tests to manual testers.

Assigning Tests to Manual Testers

Now we are going to assign the tests in our testing cycle to manual testers.

1. Click **Test Assignment >>>** on the top right. The **Test Assignment** screen appears.
2. In the **Testing Cycles** area on the left side, click on the **No specific tester** row in our testing cycle. All tests that are not assigned to a tester appear in the **Assigned Tests** area on the right side.
3. Select one or more tests, drag them to the **Testing Cycles** area on the left side and drop them to a manual tester. When assigning tests to a manual tester, the resource indicator is updated. Move the mouse over this indicator to see detailed information about the capacity, the planned time, and the time left.
4. When you are finished assigning tests to manual testers, click ... > **Start Testing Cycle** to start the tests.

When the testing cycle is started, each manual tester is informed through email about the tests that are assigned. While the testing cycle is running, a tester will also receive an email when a new test is assigned to the tester or when a test from another tester is re-assigned to the tester. You can add a test inbox, (**Manual Tests Assigned to Me**), to your personal dashboard. All the tests that are assigned to you are listed there.

Executing Manual Tests

Now that you have started your execution plan, you are ready to execute your manual test(s) using the **Manual Testing** window.

To open the **Manual Testing** window, click **Home > My Dashboard** in the menu, and then click  (**Continue Manual Test**) in the panel **Manual Tests Assigned to Me**. If this panel does not display on your dashboard, you have to add it. For more information on the dashboard and on panels, see [Dashboard](#).

The **Manual Testing** window is intuitive to use, but if you require more information on manual testing, see [Manual Testing](#).

Planning and Executing Business Process Tests

This walkthrough describes how you can plan and execute a business process test.

The assumptions for the example include:

- The business process test exists. For additional information, see [Creating a Business Process Test](#).
- The business process test includes all necessary business components and flows. For additional information, see [Adding Components and Flows to a Business Process Test](#).
- All business components in the business process test are assigned to manual testers.
- The business process test is added to an execution plan.
- The execution plan is started.

Executing Business Process Tests

Once you have started your execution plan, you are ready to execute your business process tests.

1. Open the **Manual Testing** window.

To open the **Manual Testing** window, click **Home > My Dashboard** in the menu, and then click  (**Continue Manual Test**) in the panel **Manual Tests Assigned to Me**. If this panel does not display on your dashboard, you have to add it. For more information on the dashboard and on panels, see [Dashboard](#).

The **Manual Testing** window displays the Business Process Test in the same section where manual tests are shown. However, in contrary to manual tests, a Business Process Test contains an additional area where the currently executed business component is shown. The steps section represents the steps for the current business component. For additional information on the window, see [Manual Testing Window](#).

2. Execute the steps in the steps section.
3. *Optional:* Specify values for any output parameters of the current business component.

All parameters for the current component and their value are listed after the component description.

Any input parameters that are using the `#{name}` syntax are replaced on execution with their current values.

The output parameters are listed below the input parameters with an input field to specify a value. If no value is specified, the user is warned when finishing the component. However, it is not mandatory to set a value to an output parameter when finishing a component.

For additional information on how to specify and map parameters within components and flows, see the following topics:

- [Specifying Parameters for Business Components](#)
- [Specifying Parameters for Flows](#)
- [Mapping Parameters for Business Process Testing](#)

4. When you have executed all test steps for the business component, click **Finish Component**. The component is marked as finished and the execution of the next available component of the Business Process Test is started.



Note: You can set the status of the business component to finished at any time, however, once the component is finished, you cannot change the status of the component anymore.

5. Continue until all components that are assigned to you are finished.



Note: You can also execute business process tests from the current run tab of the execution plan or from the **Activities** page.

Tracking and Analyzing Test Results - Quick Start Task

There are several ways to analyze test results in Silk Central, depending on the information you are looking for and also on the test types you are executing.

To find out more about Silk Central's results analysis features, including reporting and issue and activity tracking, see [Issue Tracking and Results Analysis](#).

Analyzing Automated Test Results

The **Test Results** view and the **Run Comparison** view provide an easy to read results overview with full drill-down capabilities. All information is included, like result files, thumbnail screenshots, messages, and stacktraces. This allows you to quickly analyze the root cause of failed test runs, especially because you can also compare the results of multiple runs side-by-side.

For example, you may be interested in performing the following tasks:

- Analyze the runs of all configurations within a configuration suite or within execution plans which represent configurations

- Compare runs of the same execution plan, for example to find out if any test statuses have changed
- Show all failed tests of an execution plan at once

To analyze the results of automated test runs:

1. In the execution **Runs** grid or the **Activities > Last Executions** grid, select one or more executions.
2. Click the **View automated test results** icon to analyze the results of a single execution, or right-click your selection and choose **Compare Automated Test Results** to compare multiple executions.
3. Click the icon at the top of the page or a specific status icon to access detailed information like result files, thumbnail screenshots, messages, and stacktraces.



Tip: Use the **Show failed tests only** check box to narrow down the displayed results.

Analyzing Manual Test Results

When you have a test run in either a finished or unfinished context, use the **Manual Test Results** view to provide an easy to read result view for manual tests including all information like attached issues and attachments. All of this information is immediately visible. You can print and download a PDF version as well.

The **Manual Test Results** view contains a tree view on the left and a details view on the right. The tree view is based on context of the selection in which the result view was opened. For example, if you selected a test cycle, you could see all the assigned testers and their tests. Selecting a specific node updates the details view with the appropriate information of the selection.

To open the **Manual Test Results** view, click the **View Manual Test Results** icon () in a **Runs** grid, in a dashboard panel, or in one of the activities pages.

Dashboard Panels

The dashboard in Silk Central is your personal starting area. You can customize it by adding and arranging various panels, which give you the information you need for your daily work.

For detailed information about dashboard panels, see [Dashboard](#).

Reports

For detailed information about reports, see [Reports](#).

Requirements

This section explains how to manage requirements in Silk Central.

A requirement is a condition or capability that a user needs to achieve a specific task, which must be met by the application under test (AUT). Associate your requirements and tests to enable measuring how well your software covers the specified requirements. Multiple tests can cover a single requirement and multiple requirements can be covered by a single test. Integrating external requirements management enables you to coordinate the requirements management features of Silk Central with other tools you may already be working with. The **Requirements** area enables you to manage and control the system requirements during development. You can create, modify, and delete requirements, associate tests with requirements, track the changes in the history, and generate tests directly from requirement lists.

Configuring Requirement Settings and Integrations

If you plan to manage your requirements in Silk Central, you can configure requirement properties, filters and change notification settings. If you are managing your requirements in another system, you can integrate it with Silk Central to make use of the requirements-management features of Silk Central, like requirement and test coverage.

Requirement Properties

The following sections describe the two types of requirement properties that are available.

Requirement Properties Page

Project:<Project Name> > **Project Settings** > **Requirement Properties**

The **Requirement Properties** page lists the custom requirement properties and the calculated properties that are available for the active project.

Calculated Properties

| Column | Description |
|----------------|---|
| Actions | Available actions that can be performed on the calculated property. |
| Name | Name of the calculated property. Click the Edit icon to edit the calculated property. |
| Formula | Contains the actual formula of the calculated property. |
| Classification | Lists the classifications for the calculated property. |
| Created On | When the property was created. |
| Created By | User who created the property. |
| Changed On | When the property was most recently modified. |
| Changed By | User who most recently modified the property. |

Custom Properties

| Column | Description |
|------------|---|
| Actions | Available actions that can be performed on the custom property. |
| Name | Name of the requirement property. Click the Edit icon to edit the requirement property. |
| Type | Property type. See the <i>Test Attribute and Requirement Property Types</i> topic for descriptions of each type of attribute. |
| Required | Shows whether the property is required. Required custom properties must be set when creating a new requirement. |
| Created On | When the property was created. |
| Created By | User who created the property. |
| Changed On | When the property was most recently modified. |
| Changed By | User who most recently modified the property. |

Calculated Properties

Calculated properties are used to create a single value based on a user-defined formula derived from the values of internal or custom properties. This value is then put into a labelled range called a *classification*. The goal of a calculated property is to be able to view the classification **Name** in grids. The value of a calculated property is shown everywhere where the values of custom properties are shown (Document view, for example). Calculated properties can also be used in filters and quality goals like other internal or custom requirement properties.

Every calculated property has a default classification without a lower bound. All other classifications must have a lower bound. The classification allows you to set a name and range for the value of the calculated property. The name of the classification is how the value will display in the grids. If one or more property values used in the formula are not set, or evaluating the formula on a particular requirement fails (for example, division by zero), --- will appear as the value of the calculated property.

You can use any internal or custom property from within a project in the formula for a calculated property. The internal or custom property must be of type `Number`, `Text`, or `List` with single selection. You can use the following operators in the formula: `+` `-` `*` `/` `()`

 **Restriction:** You can have up to five calculated properties per project. Calculated properties cannot be used within other calculated properties.

Creating a Calculated Property

Calculated properties are used to create a property that is based on a range of values. This procedure will walk you through creating a calculated property named `Custom Risk` with three classifications: `Low`, `Medium`, and `High`. We will use the internal property **Risk** and a custom property named **Impact** of type `Number` (you should create this one in advance).

1. Click **Requirements > Requirement Properties**.
2. Click **New Calculated Property**.
3. In the **Name** field, type `Custom Risk`.
4. In the **Description** field, type `Calculated property using Risk and Impact (L, M, H)`.
5. Select `Business Impact` from the **Insert Property** list. `Business Impact` is added to the **Formula** field.
6. Select `*` from the **Insert Operator** list. `Business Impact`* is added to the **Formula** field.
7. Select `Business Risk` from the **Insert Property** list. `Business Impact`*`Business Risk` is added to the **Formula** field.

8. On the default row of the **Classification** grid, rename the entry to `Low`. Since all calculated properties must contain a default classification without a lower bound, this one will be used.
9. Click **Add Classification**. A new row is added to the **Classification** grid.
10. Name the classification `Medium` and add a **Lower Bound** of 10.
11. Click **Add Classification**. A new row is added to the **Classification** grid.
12. Name the classification `High` and add a **Lower Bound** of 20.
13. Click **OK**.

A calculated property is now created that will be populated with `Low`, `Medium`, or `High` when both the **Impact** and **Risk** properties are set for a requirement.

Calculated Property Page

The **Calculated Property** page is used to add or to edit a calculated property to a project. To access this page click **Requirements > Requirement Properties** and then click an existing calculated property or **New Calculated Property**.

| | |
|-----------------------------|---|
| Name | Contains the name of your calculated property. |
| Description | Enter a meaningful description of your calculated property perhaps referencing the formula or classification. |
| Formula | Contains the formula for your calculated property. You can manually type in the formula or use the Insert Property and Insert Operator lists to do this. If you manually type in the formula, the field names should be contained within the following structure: <code>{ }</code> . Operators go between this structure. |
| Insert Property | Place your cursor where you want the property to appear in the Formula field and select one of the system or custom properties from the list to add it. |
| Insert Operator | Place your cursor where you want the operator to appear in the Formula field and select an operator. |
| Classifications Grid | The Classifications grid is used to set varying ranges for the values of your calculated properties. Click Add Classification to create a new row in the grid. Use the Name field to identify the classification and the Lower Bound field to determine the lowest value for calculated property. The important thing to know about classifications is that the Name of the classification is what will appear in grids, not the values of calculated properties. For example if your classification Name is <code>Medium</code> with a Lower Bound of 5 and your calculated property value is 7, as long as your next classification lower bound value is greater than 7, the calculated value will appear as <code>Medium</code> . |

Internal Property Values

The values of several internal properties can be used in the formulas for calculated properties. The values for supported properties are noted below.

Risk

| Value Name | Internal Value |
|------------|----------------|
| Low | 1 |
| Medium | 2 |
| High | 3 |
| Critical | 4 |

Priority

| Value Name | Internal Value |
|------------|----------------|
| Low | 1 |
| Medium | 2 |
| High | 3 |
| Critical | 4 |

Reviewed

| Value Name | Internal Value |
|------------|----------------|
| No | 0 |
| Yes | 1 |

Custom Requirement Properties

You can add custom property fields across all requirements in the selected project in **Requirements > Requirement Properties**. Custom properties can subsequently be edited alongside the default properties on the **Edit Requirements** dialog box. Custom properties are displayed in **Requirements > Details View > Properties**. If the only value of a property is a URL, Silk Central automatically displays it as a link, which enables direct access to other web applications from within Silk Central.

Creating a Custom Requirement Property

To create a new custom requirement property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirement Properties** tab.
3. Click **New Requirement Property** to display the **New Requirement Property** dialog box.
4. Type a **Name** for the new property and optionally a **Description**.
5. Select the data **Type** of the new property from the **Type** list. See the *Custom Attribute and Requirement Property Types* topic for descriptions of each type of attribute.
6. Optionally check the **Required** check box. Required custom properties must be set when creating a new requirement. On the **New Requirement** dialog, required properties that are not yet set display with an orange background and a red border.



Note: Existing requirements are not affected. But when you edit existing requirements, newly created required custom properties must be set as well to be able to save the requirement.



Note: Child requirements do not inherit required properties. If desired, these must be set explicitly for child requirements as well.

7. Click **OK**.

Your custom property is available to all requirements in the active project.

Types for Attributes and Custom Requirements Properties

Silk Central supports the following types of attributes and custom requirement properties:

Date The **Date** type supports date values.

List The **List** type supports single select or multi-select lists. It has the following controls:

- **New Item:** Allows you to create entries in the list. Click to show the **New Item** dialog box. Enter the **Name** and **Numeric Weight** for the list item and click **OK**.
 -  **Note:** The value of the **Numeric Weight** field is used during filtering when the filter criteria uses any of the following operators: **>**, **>=**, **<**, or **<=**. When the filter criteria uses the **=** or **not** operators, the value from the **Name** field is used for comparisons.
- **Order:** Use the fields in this column to sort your attributes.
- **Allow multi-select:** Click this check box to allow users to select more than one values from the list.
 -  **Note:** You can update an existing single-select list to be a multi-select list. A multi-select list cannot be changed to being a single-select list though.
- **Actions:** Use the icons in this row to delete or to edit each item in the list.

Global List

The **Global List** type allows you to reuse items from a centrally managed **List of Values**. A global list lets you maintain a set of items in a single place and use them in multiple projects. These items can be managed (added, removed, or modified) on the **List Of Values Configuration Page**. All changes on the **List Of Values Configuration Page** affect all related **Global List** attributes and custom properties in all projects.

- **Allow multi-select:** Check this check box to allow users to select more than one item from the list.
 -  **Note:** You can turn a single-select list into a multi-select list. But you can not turn a multi-select list into a single-select list.
 -  **Note:** When you disable a value from a **List Of Values**, the corresponding **Global List** item remains where it is already used but does not appear any longer in the choice of items in **Edit** operations.

A List of Values can be grouped by another List of Values (the grouping is optional). In such a case, the attributes or custom properties are connected to each other. They are added, edited, and removed coherently. This means ...

- If you add one value, you need to add the other value too.
- If you edit one value, you need to edit the other value too.
- If you remove one value, you need to remove the other value too.

User List

The **User List** type supports single-select or multi-select user lists.

Number

The **Number** type supports integer or decimal numeric values. It has the following fields:

- **Minimum value:** represents the smallest value that you can enter into the field.
- **Maximum value:** represents the largest value that you can enter into the field.
- **Decimal places:** used to define how many values to the right of the decimal point are valid. Select 0 for an integer value or 1, 2, 3, or 4 for a decimal value.

 **Note:** Attributes and custom requirement properties display their current value if an update is made to their definition (for example, changes to **Minimum value** or **Maximum value**). To apply the new definition, open and save the asset.

Text

The **Text** type supports any alphanumeric value. It has the following fields:

- **Maximum length:** represents the maximum number of characters that can be entered into the field. Enter 0 or leave this field empty for unrestricted length (for requirement properties only).

Editing Custom Requirement Properties

To edit a previously created custom requirement property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirement Properties** tab.
3. Click the name of the property you want to edit. The **Edit Requirement Property** dialog box displays.
4. Edit the name of the property in the **Name** field.
When the property is available for use, the name will display in lists for filters and requirements. Properties can be used in global filters for filtering by requirement properties and can be applied to requirements.
5. Edit the **Description** of the property.
6. See the *Custom Attribute and Requirement Property Types* topic for descriptions of each type of requirement property. You have different editing options depending on the property data type.
7. Optionally check the **Required** check box. Required custom properties must be set when creating a new requirement. On the **New Requirement** dialog, required properties that are not yet set display with an orange background and a red border.



Note: Existing requirements are not affected. But when you edit existing requirements, newly created required custom properties must be set as well to be able to save the requirement.



Note: Child requirements do not inherit required properties. If desired, these must be set explicitly for child requirements as well.

8. Click **OK** to save your changes.

Deleting Custom Requirement Properties

To delete a previously created custom requirement property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirement Properties** tab.
3. Click **X** . A confirmation dialog box appears.
4. Click **Yes** to confirm the deletion.

Requirements Management Tools Integration



Important: Before upgrading from Silk Central 15.0 or earlier to the current version, perform a **Synchronize All** to synchronize all your requirements with your integrated requirements management tools.

Integrating external requirements management enables you to coordinate the requirements management features of Silk Central with other tools you may already be working with. Integration is configured on the **Requirements Management** page in **Project:<Project Name> > Project Settings**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

Silk Central supports integration with external requirements management systems (RMS) through its open interface. Creating a plug-in and integrating it into Silk Central allows integrating any RMS. Refer to the *Silk Central API Help* for information about the interfaces that enable proper integration of external RMS.

Synchronizing Requirements

Enabling synchronization of requirements between Silk Central and an external RMS enables Silk Central to receive changes that occur in the external RMS whenever a synchronization is executed. If a project has external RMS integration enabled, the master system for requirements is automatically the external

system. This means that synchronization is always from the external RMS tool to Silk Central. Requirements can no longer be edited in Silk Central.

Property mapping functionality allows you to map property fields between Silk Central and external requirement tools. For example, a custom field in Silk Central called `user` might be equivalent to a property field in Caliber called `Field_2`. The property mapping feature ensures that changes to requirement-property fields are accurately refreshed between projects. If you don't use the property mapping feature, only the name and the description of the requirements are mapped. For more information, see *Editing Property Mapping*.

Requirements can be synchronized in one of several ways:

Manual synchronization

Click **Synchronize Changes** at the root folder level on the **Properties** page to synchronize only requirements that have changed since the last synchronization.

Forced manual synchronization

Click **Synchronize All** at the root folder level on the **Properties** page to force synchronization of all requirements.

Automatic scheduled synchronization

Based on globally defined Silk Central schedules.

Automatic online synchronization

Changes to requirements are automatically propagated between tools. This option is available for Caliber only. It requires Caliber client installation on the application server and MPX enabled on the Caliber Server (refer to the Caliber Installation Guide for detailed information on enabling MPX). To enable automatic synchronization between Caliber and Silk Central, add the file `ss.jar` to the application server at `Program Files\Silk\Silk Central <version>\lib` and restart the application server. Contact customer care to obtain the file. Requirement data is automatically updated in Silk Central when changes are made in Caliber and traces in Caliber are updated when test assignment changes are performed in Silk Central. This type of online synchronization is only available when projects are configured with the current baseline.

Automatic synchronization of requirements between Silk Central and external requirements management tools can be configured to occur based on global schedules. For details on configuring global schedules, see the *Administration* topics in this Help.



Note: The **Open Caliber** buttons open whatever program is registered as the default program for opening files of extension `.crm`. On some machines, this may be the requirement viewer, rather than Caliber. This behavior can be changed by your administrator. The client program is called `caliberrm.exe`. When properly configured, the program opens to the requirement that is selected in Silk Central.

The binder icon on the project node of the **Requirements** tree indicates the status of RM integration for the project:

No configuration

RM integration is not available.

Manual configuration

Requirement import, upload, and synchronization can be done only by clicking the corresponding buttons on the project node in **Requirements** >  > **Properties**.

At the project level, the **Properties** page includes the following properties:

| Property | Description |
|--------------|--|
| System | The external tool with which integration has been enabled. |
| Status | Whether or not integration has been enabled. |
| Project Name | The name of the external project that the Silk Central project is associated with. |

| Property | Description |
|-----------------------------|---|
| Requirement Types | The requirement types that are shared between projects. |
| Last Synchronization | Date and time of the last synchronization. |
| Last Synchronization Status | The status of the last synchronization including the number of created, updated, and deleted items. |

 **Note:** When integration between Caliber and Silk Central with automatic online synchronization has been enabled, the project node displays the current status of the online requirements change listener. The three possible statuses for such projects are: *Connected* (synchronized), *Reconnected* (synchronization recommended), and *Disconnected*.

Synchronizing Requirements Across Tools

 **Note:** Use the property mapping feature to map property fields. If you don't use the property mapping feature, only the name and the description of the requirements are mapped. For more information, see *Editing Property Mapping*.

To synchronize requirements between Silk Central and an externally configured requirements management tool

1. In the menu, click **Requirements > Details View** .
2. Select the Project node in the **Requirements** tree.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **Synchronize Changes**.
5. Click **Yes** on the **Synchronize Requirements** confirmation dialog box to begin synchronization. A dialog box opens when synchronization is complete, displaying synchronization statistics, including the number of requirements that have been created, updated, and deleted.
6. Click **OK** to complete the synchronization. Any updates that were made to mapped requirements in your externally configured requirements management tool are now reflected in the **Requirements** tree.

Synchronizing Requirements based on Schedules

You can configure global schedules to automatically synchronize requirements between Silk Central and an external requirements management tool. To enable automatic synchronization between Caliber and Silk Central, add the file `ss.jar` to the application server at `Program Files\Silk\Silk Central <version>\lib` and restart the application server. To obtain the file, contact customer care.

To synchronize requirements based on globally defined schedules:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirements Management** tab.
3. Click **Edit Schedules**. The **Edit Schedules** dialog box opens.
4. Select a predefined global schedule from the **Synchronize changes** list to define when changes in the external system should be synchronized with Silk Central.
5. Select a predefined global schedule from the **Synchronize all** list to define when all requirements in the external system should be synchronized with Silk Central.
6. Click **OK**.

For information on creating global schedules, see [Adding Global Schedules](#).

Defining Email Notification for Automatic Synchronization Events

You can define an email notification to alert users when errors occur during automated synchronization of requirements between Silk Central and external requirements management tools. All notification recipients receive copies of the synchronization log files.

To define an email notification for automatic synchronization events:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirements Management** tab.
3. Click **Edit Notification**. The **Edit Notification** dialog box displays.
4. Check the **Enable notification** check box.
5. Select a user name from the **Username** list box.
6. If required, add additional email addresses for other recipients in the **Other email addresses** field. Use semicolons to separate multiple email addresses.
7. Click **OK**.

Atlassian Jira Software Integration

Silk Central offers a requirements integration with Atlassian Jira Software, which is based on the project and issue tracking system Jira. All issue types (the generic object type in Jira) can be synchronized. Since Atlassian Jira Software offers the creation of custom issue types, you can specify a list of Atlassian Jira Software issue types in Silk Central that you want to synchronize, such as for example `Bug` or `Improvement`.

Atlassian Jira Software versions that are currently supported for integration with Silk Central:

- Atlassian Jira Software 6, 7, 8. Versions 6 and 7 are supported but not tested.
- Atlassian Jira Software Cloud. Not supported are *next-gen* projects and projects residing in a cloud instance containing *next-gen* projects.

Limitations: Synchronizing Jira requirements only works for requirements of the configured Jira project. If requirements are linked to requirements (epics) which are located in a different Jira project, these requirements will not be synchronized.

Enabling Integration with Atlassian Jira Software

To enable integration with Atlassian Jira Software:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **Atlassian Jira Software** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Enter the **URL** of your Atlassian Jira Software server.
9. Type a valid **Username** and **Password**. The user name needs to be a valid Jira user profile name.

These credentials will be used to access your Jira issue tracking system.



Important: Use the credentials of a Jira account with sufficient permissions, as missing permissions in Jira can cause inconsistencies when the plug-in attempts to add or update issues

through this account. The Jira account requires the following permissions for each project that is used in Silk Central:

- Browse Projects
- Create Issues
- Assignable User
- Modify Reporter

Additionally, the Jira account needs to be able to access (log in to) your Jira application.

Use the following credentials to integrate with Jira cloud:

- **Username:** A Jira user profile name.
- **Password:** A Jira API token. For more information on how to generate the API token, refer to the Jira documentation.

Use the following credentials to integrate with an on-premise Jira instance:

- **Username:** A Jira user profile name.
- **Password:** A Jira password.

10. In the **Issue Types** field, specify the issue types that you want to synchronize, separated by semicolons (;).

All issue types (the generic object type in Jira) can be synchronized. Since Atlassian Jira Software offers the creation of custom issue types, you can specify a list of Atlassian Jira Software issue types in Silk Central that you want to synchronize, such as for example `Bug` or `Improvement`.

11. *Optional:* To limit the amount of requirements that are being synchronized, you can use the **Custom JQL** field to pre-filter requirements before they are being synchronized. Create a query using Jira Query Language (JQL) within your Jira system, then copy and paste the string into the **Custom JQL** field. For more information on JQL, visit Atlassian's [Advanced Searching Documentation](#).

 **Tip:**

- Do not include `ORDER BY` statements in your JQL query.
- The JQL string may not exceed 2000 characters.
- The JQL query is used additionally to your configured project and issue types, it does not overrule those settings.

12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.

A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

13. Click **Load**. The **Project** list is populated with all the projects in Jira that you have permissions for.

14. From the **Project** list, select the external project with which the Silk Central project is to be integrated.

15. Click **OK** to save your settings.

Atlassian Jira Software Property Mappings

The following properties have a default mapping between Jira and Silk Central:

- User stories that are in Jira's `Backlog` are reflected in Silk Central by having the `Sprint` property value set to `Backlog`. This enables you to easily search for them by creating a filter.
- The default `Priority` values of Jira's user stories are mapped to Silk Central's `Priority` property. If you are not using Jira's default `Priority` values, you can customize this mapping.

Viewing Jira Backlog Stories in Silk Central

1. In the menu, click **Requirements > Details View**.

2. Create a new filter with the following criteria:
 - **Selection Criteria:** External Custom Attributes
 - **Property:** Sprint
 - **Value:** Backlog
3. Click **Apply** to apply the filter immediately or **Save and apply** to save the filter for future reuse.

Mapping Jira Priority values to Silk Central's Priority

The default `Priority` values of Jira's user stories are mapped to Silk Central's `Priority` property as follows:

- Blocker=SC_Priority_Critical
- Critical=SC_Priority_High
- Major=SC_Priority_Medium
- Minor=SC_Priority_Low
- Trivial=SC_Priority_Low

If you are not using Jira's default `Priority` values, or if you want to customize the mapping, your Silk Central administrator should customize the settings as follows:

1. Stop the application server.
2. Open the file `propertyMappings.properties` with a text editor. This file is located in the Jira plugin-file `JIRA Agile.zip` on the application server in the `plugins` folder.
3. Search for `SC_Priority=Priority`, beneath which you can customize the mapping of Jira's `Priority` values to Silk Central's `Priority`.



Note: Silk Central `Priority` values require the prefix `SC_Priority_`.

4. Save and close the file
5. Re-start the application server.

CA Agile Central Integration

Integrating the project management tool CA Agile Central enables you to define user stories which you can then use as requirements in Silk Central. You can then create tests in Silk Central to cover these requirements. To update the tests you have created with any changes to the corresponding user stories in CA Agile Central, synchronize the requirements.

The following user story properties are always synchronized from the CA Agile Central REST Service. Custom fields are also synchronized if they are specified in the integration configuration. You can use these properties to help filter and categorize the user stories.:

| Property | Description |
|----------------------------|--|
| Name | The name of the user story in CA Agile Central. |
| CA Agile Central ID | The identifier of the user story in CA Agile Central. This property is the External ID of the requirement in Silk Central. |
| Description | The description of the user story in CA Agile Central. |
| Iteration | The iteration in CA Agile Central in which the user story is included. |
| Project | The project in CA Agile Central in which the user story is included. |
| Release | The project release in which the user story is included. |
| State | The state of progress of the user story in CA Agile Central. The following states are available: <ul style="list-style-type: none"> • Defined |

| Property | Description |
|-------------------------------------|---|
| | <ul style="list-style-type: none"> • In-Progress • Completed • Accepted |
| Plan Estimate (Story Points) | The estimated time in CA Agile Central for the user story to be completed. The time is estimated in story points. |

Enabling Integration with CA Agile Central

To enable integration with CA Agile Central:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.

 **Note:** You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **CA Agile Central** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Type the **URL**.
9. Type valid **Username** and **Password** credentials for CA Agile Central, or type a valid **API Key** instead. An API Key can be used to access your CA Agile Central subscription data without using your username and password. For more information about CA Agile Central API Keys, refer to the [CA Agile Central Application Manager page](#).
10. *Optional:* If direct access to the Internet is restricted, and the CA Agile Central REST service is located in a machine outside the scope of the internal network, use the **Proxy Host** and **Proxy Port** fields to specify a proxy through which Silk Central can connect to CA Agile Central.
 **Note:** To connect to CA Agile Central by using a proxy, you need to fill out both fields, **Proxy Host** and **Proxy Port**.
11. *Optional:* If you want to synchronize custom fields, type their names into the **Custom Fields** field, separated by commas. The internal field names must be used, not the display names.
12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct. A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.
 **Note:** Consult your system administrator if you are not able to establish a connection.
13. Click **OK**.
14. Click **Load**. The **Project** list box is populated with all the projects from all workspaces in CA Agile Central that you have permissions for.
15. From the **Project** list box, select the external project with which the Silk Central project is to be integrated.
16. Click **OK** to save your settings.

Caliber Integration

This section describes how to integrate Caliber with Silk Central.

Configuring integration with Caliber requires the definition of Caliber login credentials. Whenever requirements are synchronized between Silk Central and Caliber, these credentials are used to login to Caliber, thus checking out a Caliber license. The license is set free as soon as the synchronization process has completed. We recommend creating a dedicated Caliber user for synchronization purposes, which should be used by all Silk Central requirements integrations with Caliber. This ensures that only a single Caliber license is used for the process of synchronization.

Caliber versions that are currently supported for integration with Silk Central: Caliber 11.4, 11.5

 **Note:** Caliber user defined attribute **Multiple selection user list** and **Multiple selection group list** must be mapped to the Silk Central text attribute type and not list type.

Enabling Integration with Caliber

To enable integration with Caliber:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.

 **Note:** You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **Caliber** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Type the **Hostname** of the machine where the external server is installed.
9. Type valid **Username** and **Password** credentials for the requirements management server.
10. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.
A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.

 **Note:** Consult your system administrator if you are not able to establish a connection.

11. Click **OK**.
12. From the **Project** list box, select the external project with which the Silk Central project is to be integrated. The requirement types that are available with the selected project are automatically populated into the **Requirement Types** field. The baselines that are available with the selected project are automatically populated into the **Baseline** field.
13. Select a **Baseline** from the external project that should be integrated with the Silk Central project.
Your selections are displayed on the **Edit Configuration** dialog box.
14. Click **OK**.
15. Select one or more requirement types from the external project that should be integrated with the Silk Central project.
Use **Ctrl+Click** to select multiple requirement types.
16. Click **OK** to save your settings.

Baseline Support for Caliber Integration

You can select the current baseline or existing user-defined baselines for Caliber integration. You cannot import modified baselined requirements into Silk Central. Requirements that are not of the current baseline can only be changed in Caliber if the version of the requirement that is used for the baseline is changed. Such changes are only updated within Silk Central requirements when a manual or scheduled synchronization is performed.

You can change a baseline after you import it into Silk Central. You can change the configured baseline to a different user-defined baseline or the current baseline. After such a change, the next synchronization of the baseline, either manual or scheduled, will update the Silk Central project and update, create, or delete requirements as required. When a baseline is changed, a message displays that states that the changes will take effect after the next synchronization.

Handling Test Assignments in Caliber

Assigned tests are displayed, managed, and created as traces ("Trace to") of synchronized requirements in Caliber.

The External Traceability for Silk Central must be enabled for a Caliber project within the Caliber Administrator and the correct Silk Central front-end server must be configured. Click **Edit** for the Silk Central External Traceability for correct project within the Caliber Administrator.

When a project is synchronized with a Silk Central project, the assigned tests of the synchronized Silk Central requirements will display as traces ("Trace To") of the Caliber requirement. If you edit these assignments in Silk Central, the changes will be reflected immediately in Caliber.

Copying Caliber-Integrated Projects

To manage Caliber baselines when copying Silk Central projects:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirements Management** tab.
3. Verify that the baseline you want to save is selected.



Note: If a baseline is changed, you must perform a synchronization to update the project requirements with the baseline changes, before you can copy an associated Silk Central project. The integration configuration is only copied if you select a baseline other than the current baseline. If you select the current baseline, you need to specify if you want to keep the integration configuration in the original project or move it to the copied project.

4. If the baseline that you want to save is not selected, click **Edit Configuration**. The **Edit Configuration** dialog box displays.
5. Click **Browse** next to the **Project name** field. The **Browse Projects** dialog box opens.
6. Select the baseline you want to save, then confirm your selection.
7. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
8. Click  in the **Actions** column of the project you want to copy.



Note: For full details on copying a project, see the *Administration* topics in this Help.

The **Copy Project** dialog box displays.

9. Select the items you want to copy into the new project, then confirm your selection.
10. Apply the baseline that you want to continue working with to the Silk Central project.



Note: After copying a project, the original project and the copy are identical. Define on which you will continue working on by applying the corresponding baseline.

Creating a Baseline for a Caliber-Integrated Project

To create a baseline of a Caliber-integrated Silk Central project:

1. Create a baseline of the Silk Central project. A Silk Central message box displays during the process, asking if you want to enable the Caliber integration for the new baseline.
2. Click **Yes**. When the new baseline is created, the **Baseline Project - Adapt Project Settings** dialog box opens.
3. Click **Edit** to change the requirements integration settings.
The **Edit Configuration** dialog box opens.
4. Click **Browse** next to the **Project name** field. The **Browse Projects** dialog box opens.
5. Select a Caliber project, a baseline, and the requirement type for synchronization.
6. Click **Ok** to close the **Browse Projects** dialog box.
7. Click **Ok** to close the **Edit Configuration** dialog box.
8. Click **Finish** to close the **Baseline Project - Adapt Project Settings** dialog box.

Generating Tests from Scenarios and Simulations

Generate tests from the Caliber objects **Scenario** and **Simulation**.

To generate tests from the Caliber objects **Scenario** and **Simulation**:

1. Establish integration with Caliber.
For additional information, see *Enabling Integration with Caliber*.
2. Select the **Scenario** or **Simulation** in the **Requirements** tree from which you wish to generate tests.
3. On the **Properties** tab, click **Generate Tests**. The **Generate Tests** dialog box displays.
4. Use **Ctrl+Click** or **Shift+Click** to select the tests that you want to generate.
5. Click **Generate All** to generate all tests, or click **Generate Selected** to generate the selected tests. The **Tests** tree displays.
6. Select the folder to which you want to add the generated tests.



Note: Tests that already exist in the selected destination folder are not created, but updated. No tests in the folder are deleted.

7. Click **OK**.

IBM Rational DOORS Integration

This section describes how to integrate Silk Central and IBM Rational DOORS. Silk Central 20.6 offers out-of-the-box support for IBM Rational DOORS, without the need to customize anything in your DOORS installation. If you are upgrading from Silk Central 15.5 or earlier though, and using customized dx1 files, you will need to overwrite the dx1 files in the `lib\dx1` folder within your Silk Central installation folder with your customized dx1 files.

IBM Rational DOORS versions that are currently supported for integration with Silk Central: IBM Rational DOORS 9.5, 9.6

Enabling Integration with IBM Rational DOORS

To integrate Silk Central and DOORS, install the DOORS client on the Silk Central front-end server machine. If you use more than one front-end server machines, you must install the DOORS client on each front-end server that you have configured for integration with DOORS.

To enable integration with DOORS:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.

3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **IBM Rational DOORS** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. In the **RM service URL** field, type the URL of the Silk Central DOORS requirement Web Service.
The default value should point to the correct location already. For example `http://MySCTMHost:19120/services/doorsrequirementsmanagement`.
9. Type valid **Username** and **Password** credentials for the requirements management server.
10. The default DOORS client installation path is displayed in the **DOORS Installation Path** field. If this path is not correct, enter the correct destination within the front-end server directory structure.
11. *Optional:* To synchronize only specific requirements, based on a filter that has been specified and activated for a view in DOORS, specify the name of the view in the **DOORS Filter View** field.
12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.
A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: If **Test Connection** fails with the error 80004005 Unspecified error: Access denied: Doors session is not authenticated, the issue might be that the DOORS setting **Prompt to open all modules from last session on startup** is enabled. To solve this issue, disable this setting in the DOORS options. Consult your system administrator if you are not able to establish a connection.

13. Click **Browse** next to the **Project name** field to open the **Browse Requirement Types** dialog box. From the **Project** field, select the external project with which the Silk Central project is to be synchronized.
The requirement types that are available with the selected project are automatically populated into the **Requirement types** field. Select the requirement types that are to be synchronized and click **OK**. Use **Ctrl+Click** to select multiple requirement types.
Your selections are now displayed on the **Edit Configuration** dialog box.
14. Click **OK** to save your settings.



Caution: As the DOORS application object is used for communication, and this object does not support login data, but rather requires a running DOORS client, Silk Central starts each DOORS client process with the provided login data and then uses that same data for all subsequent application objects. Therefore only one set of DOORS login credentials is supported for communication at one time. It is recommended that you use the same DOORS credentials for all configurations so that integration tasks can be performed on the front-end server for all projects at the same time. When a second set of credentials is used, the second set only works after all sessions using of the first set of credentials have timed out.

IBM Rational DOORS Next Generation Integration

This section describes how to integrate Silk Central and IBM Rational DOORS Next Generation. Silk Central 20.6 offers out-of-the-box support for IBM Rational DOORS Next Generation, without the need to customize anything in your DOORS installation.



Note: All artifacts of the project are synchronized, except the DOORS collection artifacts **Module** and **Collection** themselves.

IBM Rational DOORS Next Generation versions that are currently supported for integration with Silk Central: IBM Rational DOORS Next Generation 6.0

Enabling Integration with IBM Rational DOORS Next Generation

To enable integration with IBM Rational DOORS Next Generation:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **IBM Rational DOORS Next Generation** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. In the **DOORS Next Generation URL** field, type the fully qualified URL of the IBM Rational DOORS Next Generation server, for example `https://MyDOORSHost.MyDomain.com:9443/rm`.
9. In the **Authentication URL** field, type the fully qualified URL of the IBM Rational Jazz Team Server, for example `https://MyDOORSHost.MyDomain.com:9443/jts`.



Note: The new IBM Rational Authentication Server is not supported for authentication, only the IBM Rational Jazz Team Server.

10. Type valid **Username** and **Password** credentials for the IBM Rational DOORS Next Generation server.
11. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.
A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

12. Click **Load** next to the **Project name** list to populate the list with the available DOORS projects. Select the external project with which the Silk Central project is to be synchronized.
13. Click **OK** to save your settings.

SAP Solution Manager Integration

To purchase the SAP Solution Manager Plug-In, please contact your sales representative.

SAP Solution Manager facilitates technical support for distributed systems, covering solution deployment, operation, and continuous improvement. SAP Solution Manager is a centralized, robust application management and administration solution, that combines tools, content, and direct access to SAP to increase the reliability of solutions and lower total cost of ownership. For additional information on the SAP Solution Manager, refer to the SAP Solution Manager documentation.

The integration between Silk Central and SAP Solution Manager allows for the synchronization of SAP Solution Manager business processes and incidents as Silk Central requirements and issues.

System Requirements for SAP Solution Manager Integration

The following are the system requirements for the SAP Solution Manager integration plugin:

- SAP Solution Manager 7.2.
- Silk Central 19.0 or later.
- The *Silk Central Integration Hub for SAP Solution Manager* must be installed in SAP Solution Manager.

Configuring SAP Solution Manager

Configure SAP Solution Manager to enable the integration with Silk Central.

For additional information, see

Configuration_Guide_SilkCentral_Integration_Hub_for_SAP.pdf and *User_Guide_SilkCentral_Integration_Hub_for_SAP.pdf*, which are included in the SAP Solution Manager plug-in.

For this integration to get valid links from SAP Solution Manager to Silk Central, the front-end server machine of Silk Central must have a valid fully qualified name.

Adding SAP Solution Manager Issue Tracking Profiles

To add a SAP Solution Manager issue tracking profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Issue Tracking** tab. The **Issue Tracking** page opens, listing all of the issue tracking profiles that have been created for the system.
3. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Type a **Description** for the new profile.
6. Select **SAP Solution Manager** from the **Type** list box.
7. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
8. Type the **URL** of the Z-SCTM_ADAPTER web service of SAP Solution Manager.
9. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
10. Click **Load Default Incident Type** to load all possible incident types and populate the **Default Incident Type** list box.
11. Select an incident type from the **Default Incident Type** list box. If you are creating links to existing incidents in SAP Solution Manager and you do not specify the incident type, this type will be used for it.
12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

13. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Enabling Requirements Integration with SAP Solution Manager

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .

4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **SAP Solution Manager** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Enter the **URL** of the **Z_SCTM_ADAPTER** web service of SAP Solution Manager.
9. Enter the **Username** and **Password**.
10. Click **Test Connection** to confirm that the host and user credentials are correct. If the settings are correct, a **Test connection was successful** message displays.
11. Click **Load** to load all available solution and branch combinations and select the correct one.
Silk Central stores such combinations of solutions and branches as projects.
12. *Optional:* Check the **Enable creation of unassigned requirements** check box to enable creation and editing of unmapped requirements in Silk Central projects that are configured for integration.
13. Click **OK** to confirm and close the dialog.

Working with External Properties

This section explains how to work with external properties in Silk Central.

Editing External Properties

To edit external properties:

1. In the menu, click **Requirements > Details View**.
2. Select the requirement for which you intend to edit external properties.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **Edit External Properties**.

The **Edit External Properties** dialog box displays. All properties of the external requirement are displayed here. Edit all properties as required.



Note: Editable properties on this dialog box offer input fields and controls with which you can edit the properties. If a mapping rule exists for an attribute, the attribute will be tagged with a trailing asterisk (*).

5. Click **OK** to save your changes and close the dialog box.

Viewing External Properties

To view external properties:

1. In the menu, click **Requirements > Details View**.
2. Select the requirement.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **View External Properties**. The **View External Properties** dialog box displays. All properties of the external requirement are displayed here.
5. Close the dialog box.

Editing Property Mapping

The property mapping functionality allows you to map property fields between Silk Central and external requirements management tools. For example, a custom requirement property in Silk Central called `User` might be equivalent to a custom property in Caliber called `User_ID`. The property mapping feature ensures that requirement property fields are accurately populated between projects during requirement uploading and importing. If there are multiple requirement types, you must map each type separately.

 **Note:** If you don't use the property mapping feature, only the name and the description of the requirements are mapped.

To edit property mapping:

 **Note:** The following applies for synchronizing properties with boolean values: since Silk Central does not support the boolean data type, you need to create a single select list with the following two list items:

- Name = `Yes`, Numeric Weight = 1
- Name = `No`, Numeric Weight = 0

 **Note:** When you map lists, requirement properties from both products need to have the same named options for each list item, and they are case sensitive.

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Edit Property Mapping** for the configured external tool. The **Edit Property Mapping** dialog box opens.
5. Select an external requirement type from the **Requirement types** list. All custom requirements of that type are then displayed below in the selection box.
6. Select the custom requirement property for which you are establishing mapping.
7. From the list box on the right, select the Silk Central custom property to establish mapping to the external custom property you have selected.
8. Click **Add Mapping** to map the requirements. The results are displayed in the **Custom property mapping** box.
9. The **System property mapping** box displays the two pre-configured mappings for requirement name and description, which cannot be removed.
10. Click **OK** to save your changes.

Deleting Property-Mapping Value Pairs

To delete a property-mapping value pair:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Edit Property Mapping** for the configured external tool. The **Edit Property Mapping** dialog box opens.
5. Select the property-mapping value pair in the **Custom property mapping** select box.
6. Click **Remove Mapping**.
7. Click **OK** to save your changes.

Disabling Requirements Management Integration

To disable requirements management integration configuration:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click the **Disable Configuration** button of the requirements management tool for which you want to disable integration.

All integration data and functionality is disabled, but not deleted from the project.

Removing Requirements Management Integration

To remove requirements management integration:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Remove Configuration** of the requirements management tool for which you want to remove integration.

This button is only enabled if the configuration is disabled.

The **Remove External Association** dialog box displays.

5. Click **Yes**.

All related data is removed from the database.

Filters

Filters provide an efficient means of finding exactly the information you need, while excluding extraneous detail. Filters highlight only those elements that are relevant to your needs, and enable you to quickly sort through requirements, test elements, and execution plans. By defining global filters, you can create complex filter criteria that are available throughout Silk Central without defining filter criteria each time you need to filter a list.

Based on your needs, you can create new filters, edit existing filters, select filters, delete filters, or turn filtering off at the project level. Projects do not contain default filters. You can access and edit filters from the toolbars in the Silk Central units and from the **Project Settings** unit.



Note: Filters are not applied to reports. The **Recent Changes** filter enables you to view project-wide changes and additions that other users have made to tests since your last change acknowledgement. The **Show Changes/Show All** toggle button and the **Acknowledge** button in the **Tests** area help you to find out what changes other users have made. Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Creating Filters

To create a filter:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Filters** tab to view the list of available filters.
3. Click **New Filter**. The **New Filter** dialog box appears.
4. Type a **Name** for the new filter.
This name will be displayed in list boxes when the filter becomes available.
5. Select a **Category** from the list to make the filter available in the **Requirements, Tests, or Execution Planning** area of Silk Central.
6. *Optional:* Type a **Description** for the new filter.

7. *Optional:* Check the **Visible to other users** check box to allow other users to see the filter.
8. *Optional:* Check the **Editable by other users** check box to allow other users to edit the filter.

 **Note:** If **Visible to other users** and **Editable by other users** are checked, the filter is public. To delete non-public (private) filters, you have to be the owner of the filter or you need the **Delete private filters of other users** permission.

9. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.

 **Note:** You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

10. Select a **Property**, **Operator**, and **Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|--------------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

11. *Optional:* Click **More** if you want to add more than one filter category to the new filter. Repeat this procedure to define new categories.

 **Note:** If you define more than one filter category, you must define whether the categories need to be fulfilled in addition to the existing categories (AND relationship), or if the filter returns true when one of the filter categories is fulfilled (OR relationship). Select either **AND** or **OR** to define the filter category relationship. You cannot define nested AND, OR relationships.

12. *Optional:* To remove filter categories, click **Fewer**. This removes the last filter category.

13. Click **OK** to save the new filter, or click **Cancel** to abort the operation.

Creating Advanced Filters

Advanced custom filters enable you to combine simple filters to create complex filters that apply multiple filter criteria simultaneously.

To create an advanced custom filter:

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. Click  (**New Filter**) in the toolbar. The **New Filter** dialog box appears.
3. If necessary, click **Advanced** to show the whole dialog box.
4. Click **More** to display a second set of filter-parameter fields with which you can define a second set of filter parameters.

5. Select a logical operator for the application of the filtering queries. For example, with the operator `and` filtered elements must meet both sets of criteria and with the operator `or` filtered elements must meet one, but not both, of the criteria sets.
6. To delete a filter-parameter string, click .
7. To display additional filter-parameter fields and create additional filter queries, click **More**. To remove excess filter-parameter sets, click **Fewer**.

Editing Filters

To edit a filter:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Filters** tab to view the list of available filters.
3. Click the name of the filter you want to edit. The **Edit Filter** dialog box displays.
4. Edit the **Name** and **Description** of the filter.
5. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.



Note: You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

6. Select a **Property**, **Operator**, and **Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|--------------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

7. Click **OK** to save the edited filter definition.

Copying Filters

Create a new filter by copying an existing one and changing the settings.

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. In the toolbar of the selected area, select a filter from the list box.
3. Click . The **Edit Filter** dialog box opens with the settings of the original filter.
4. Edit the properties of the filter and click **OK**.

Deleting Filters

To delete a filter:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Filters** tab to view the list of available filters.
3. Click **Delete Filter** for the filter that you want to delete. A confirmation dialog box displays, asking you to confirm the deletion.
4. Click **Yes** to delete the selected filter or **No** to abort the operation. If you click **Yes**, you will be returned to the filters list and the deleted filter is no longer displayed.

Global Filter Details

Global filters allow you to quickly sort through elements in a Silk Central area, highlighting only those that are relevant to your needs.

To view the details of a global filter, click **Project:<Project Name> > Project Settings > Filters** in the menu, and click on the name of the filter.

The specific Silk Central area in which a new filter is available is defined by selecting the general filter category.

Each global filter has to have a property, an operator, and a value from the respective list boxes:

- Property** Available properties depend on the filter category. It defines the property for which you are defining a filter setting. If you selected an attribute category, the property list includes custom attributes to query against.
- Operator** Specifies the filter operator. The operator depends on the property type. For example, if you select a property that is based on a string field type, the available operators are = (equals defined value), *not* (differs from the defined value), *contains* (contains the defined value somewhere in the string), and *does not contain* (does not contain the defined value in the string).
- Value** The value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box.

Recent Changes

The *Recent Changes* filter enables you to efficiently view and acknowledge changes and additions that other users have made to requirements, tests, or execution plans project-wide since your last change acknowledgement.

The two buttons at the far-right of the toolbar, the **Show Changes/Show All** toggle button and the **Acknowledge** button, help you to find out what changes other users have made.



Note: Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Applying Filters

After you have created and stored a custom filter, you can apply that filter to the selected tree. Custom filters can be applied for requirements, tests and execution plans. Only elements that meet applied filter criteria are displayed in the tree.



Note: Filtered requirements are returned in read-only form and can not be edited. The **Edit Properties** button is disabled for filtered requirements.

To apply a stored filter:

1. In the menu, select the appropriate area: **Requirements, Tests, or Execution Planning**.
2. Select the desired filter from the **Filter** list box on the toolbar.
3. All elements that meet the filter's criteria are displayed.



Note: To remove filtering and display all elements, select **<No Filter>** from the **Filter** list box on the toolbar.

Change Notification

Silk Central can notify you by email when requirements or tests are changed by other users. Each user has the option of activating change-notification. Once notification has been enabled, perform an initial acknowledgment. From then on, you can view and acknowledge changes that have occurred since your last acknowledgment. To avoid numerous notifications, only a single email alert is sent to you when a change is made, regardless of how many changes other users may have made since your last acknowledgment. Email alerts include links that take you directly to a view of recent changes.

Before you can activate change notification for requirements or tests, you must configure your email address in the user settings of Silk Central.



Note: Change notification only works if an email server has been configured by your administrator. If change notification has not been enabled, please contact your Silk Central administrator.

Enabling Change Notification

To enable change notification:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Notifications** tab.
3. Click **Configure Email Notification** to open the **Configure Email Notification** dialog box.
4. If you want to be notified by email when changes are made to requirements in the currently selected project, check the **Changes on Requirements** check box.
5. If you want to be notified by email when changes are made to tests within the currently selected project, check the **Changes on Tests** check box.
6. Click **OK** to save the notification settings, or click **Cancel** to abort the operation without saving changes.

You will be notified by email about changes for which you have activated notification.

Disabling Change Notification

To disable change notification:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Notifications** tab.
3. Click **Configure Email Notification** to open the **Configure Email Notification** dialog box.
4. If you do not want to be notified by email when changes are made to requirements in the currently selected project, uncheck the **Changes on Requirements** check box.
5. If you do not want to be notified by email when changes are made to tests in the currently selected project, uncheck the **Changes on Tests** check box.
6. Click **OK** to save the notification settings or click **Cancel** to abort the operation without saving changes.

Changes Triggering Change Notification

When you have activated change notification to inform you of changes that are made to requirements or tests an email alert is sent to you, following your logout. The email alert is sent to you if one or more of the following settings are changed:

| Area | Changes |
|--------------|---|
| Requirements | <ul style="list-style-type: none"> • A requirement is created or deleted. • The name or description of a requirement is edited. |

| Area | Changes |
|-------|--|
| | <ul style="list-style-type: none"> • A system property is edited. • A requirement is set as obsolete. • A requirement is recovered. • A test is assigned to or removed from a requirement. • A custom property of a requirement is created, edited, or deleted. |
| Tests | <ul style="list-style-type: none"> • A container is created or edited. • A product is edited. • A source control profile is edited. • "Clear working folder" is edited. • The root node is edited. • The custom data directory is edited. • The include directory is edited. • The hidden test properties are edited. • The Silk Test Classic interface is edited. • A test folder is created, edited, or deleted. • A test is created, edited, or deleted. • The planned time is edited. • A test step is added or edited. |

Notifications Page

Project:<Project Name> > Project Settings > Notifications

The **Notifications** page lists the notification events for the active project. For each notification event, the **Notifications** page displays the following columns:

| Column | Description |
|---------------------|---|
| Notification Events | Name of the notification event that has been set up for the active project. |
| Status | Status of the notification event. Active or inactive. |

Click the **Configure...** button in each section to edit the values.

Email

When an email notification event is activated, a notification email is sent to the user that activated the event, the first time one of the specified settings is changed.



Note: You must have specified an email address to be able to receive email notifications.

Requirement Flagging

You can configure notifications differently for internal and external requirements. All requirement flagging is enabled by default. For more information, see the *Flags* topic.

Managing Requirements

Add or edit requirements to establish and maintain agreement between customer and developer on both technical and non-technical requirements.

Creating Requirements

To create a new requirement:

1. In the menu, click **Requirements > Details View** .
2. In the **Requirements** tree, select a requirement or the root node.
3. Click  (**New Requirement**) on the toolbar to create a new requirement on the same hierarchical level, or  (**New Child Requirement**) to create it hierarchically beneath the selected requirement. The **New Requirement** dialog box appears.
4. Type a **Name** and a **Description** for the requirement.
 **Note:** Silk Central supports HTML formatting and cutting and pasting of HTML content for description fields.
5. *Optional:* Uncheck the **Inherit from parent** check boxes. In this case the child requirement will not inherit the properties from the parent requirement. By default, all check boxes are checked.
 **Note:** In the **Details View** on the **Properties** page and in the **Document View**, inherited properties are marked with asterisks (*).
6. Select the appropriate **Priority**, **Risk**, and **Reviewed** status from the lists.
7. If custom requirements were defined, they are listed below the **Reviewed** list. Type in the corresponding field any custom property data that you want to track with this requirement. This is disabled for inherited properties. For details regarding creating, editing, and deleting custom requirement properties, see *Custom Requirement Properties*.
8. To finish creating a new requirement:
 - Click **OK** to create the requirement and to close the dialog box.
 - Click **OK and New Requirement** to create the requirement and to keep the dialog box open to enter data for another requirement.
 - Click **OK and New Child Requirement** to create the requirement and to keep the dialog box open to enter data for a child requirement.

Editing Requirements

To edit the properties of a requirement:

1. In the menu, click **Requirements > Details View** .
2. In the **Requirements** tree, select a requirement.
3. Click  (**Edit**) in the toolbar.
You can also right-click a requirement and click **Edit** or click the **Properties** tab and click **Edit Properties**.
The **Edit Requirement** dialog box appears.
4. Type a **Name** and a **Description** for the requirement.
 **Note:** Silk Central supports HTML formatting and cutting and pasting of HTML content for description fields.
5. *Optional:* Uncheck the **Inherit from parent** check boxes. In this case the child requirement will not inherit the properties from the parent requirement. By default, all check boxes are checked.
 **Note:** In the **Details View** on the **Properties** page and in the **Document View**, inherited properties are marked with asterisks (*).
6. Select the appropriate **Priority**, **Risk**, and **Reviewed** status from the lists.

7. If custom requirements were defined, they are listed below the **Reviewed** list. Type in the corresponding field any custom property data that you want to track with this requirement. This is disabled for inherited properties. For details regarding creating, editing, and deleting custom requirement properties, see *Custom Requirement Properties*.
8. Click **OK** to save your changes.

Marking Requirements as Obsolete

Deleting requirements destroys them permanently. Sometimes it is preferable to mark requirements as obsolete. Obsolete requirements can be hidden, shown, and recovered.

To mark a requirement as obsolete:

1. In the menu, click **Requirements > Details View** .
2. In the **Requirements** tree, select a requirement.
3. Click  (**Delete**) in the toolbar or right-click the requirement and click **Delete**. A dialog box appears.
4. Click **Yes**.



Note: The **Destroy permanently** check box is unchecked by default.



Note: Obsolete requirements are displayed in italics in the **Requirements** tree.

To show obsolete requirements, right-click any requirement in the **Requirements** tree and click **Show Obsolete Requirements**.

To hide obsolete requirements, right-click any requirement in the **Requirements** tree and click **Hide Obsolete Requirements**.

Recovering Obsolete Requirements

To recover an obsolete requirement:

1. In the menu, click **Requirements > Details View** .
2. If the obsolete requirements are hidden, right-click any requirement in the **Requirements** tree and click **Show Obsolete Requirements**.
3. Right-click on the obsolete requirement in the **Requirements** tree and click **Recover**.



Note: Obsolete requirements are displayed in italics in the **Requirements** tree.

To hide the obsolete requirements again, right-click any requirement in the **Requirements** tree and click **Hide Obsolete Requirements**.

Permanently Deleting Obsolete Requirements

To permanently delete an obsolete requirement:

1. In the menu, click **Requirements > Details View** .
2. If the obsolete requirements are hidden, right-click any requirement in the **Requirements** tree and click **Show Obsolete Requirements**.
3. In the **Requirements** tree, select an obsolete requirement.
4. Click  (**Delete**) in the toolbar or right-click the requirement and click **Delete**. A dialog box appears.
5. Click **Yes** to permanently delete the requirement.



Note: Obsolete requirements are displayed in italics in the **Requirements** tree.

To hide the obsolete requirements again, right-click any requirement in the **Requirements** tree and click **Hide Obsolete Requirements**.

Requirements Reports

Requirements reports detail the status of functional requirements, for example compatibility requirements, GUI requirements, or feature requirements, which must be met during development.

Requirements may also relate to product management objectives such as reliability, scalability, and performance. The requirement-management reports help managers to determine if adequate test coverage is established to verify that system requirements are met during development. When a report references a requirement that includes HTML-formatted content, that content is rendered in the report.

Status Reports

The following status reports are available for the **Requirements** area:

| | | | | | | | | | |
|---|---|---------------|------------------|---------------|------------------|------------|---|---------------------|--|
| Requirements Status Overview | Represents a grouped summary of all requirements by current requirement coverage. Coverage is expressed by the statuses <code>Passed</code> , <code>Failed</code> , <code>Not Executed</code> , and <code>Not Covered</code> . | | | | | | | | |
| Top-Level Requirement Coverage | Represents a listing of all top-level requirements. For each requirement the number of child requirements, that are covered and not-covered by tests, is displayed. | | | | | | | | |
| Status of Requirements with Priority Higher than 'X' | Represents a summary of all requirements by current requirement coverage. The returned group of requirements is restricted by the <code>Priority</code> parameter, which specifies the lowest requirement priority that is considered in the data. | | | | | | | | |
| Requirement Impact Analysis Report | Displays the test details grouped by the execution plan that is associated to the requirement. This report allows the user to gain insight into testing assets that may be impacted by a change to the requirement. The report contains the name, status, date/time of last execution, manual testers, planned time, and issues of each test. The following statuses are used: <table border="0" style="margin-left: 20px;"> <tr> <td>Passed</td> <td>The test passed.</td> </tr> <tr> <td>Failed</td> <td>The test failed.</td> </tr> <tr> <td>N/A</td> <td>The test is associated to a requirement but is not submitted for execution.</td> </tr> <tr> <td>Not Executed</td> <td>The test is associated to a requirement and is submitted for execution but not executed yet.</td> </tr> </table> | Passed | The test passed. | Failed | The test failed. | N/A | The test is associated to a requirement but is not submitted for execution. | Not Executed | The test is associated to a requirement and is submitted for execution but not executed yet. |
| Passed | The test passed. | | | | | | | | |
| Failed | The test failed. | | | | | | | | |
| N/A | The test is associated to a requirement but is not submitted for execution. | | | | | | | | |
| Not Executed | The test is associated to a requirement and is submitted for execution but not executed yet. | | | | | | | | |
| Project Requirement Traceability Report | Displays all requirements in a project along with their associated tests and issues. The following statuses are used: <table border="0" style="margin-left: 20px;"> <tr> <td>Passed</td> <td>The test passed.</td> </tr> <tr> <td>Failed</td> <td>The test failed.</td> </tr> <tr> <td>N/A</td> <td>The test is associated to a requirement but is not submitted for execution.</td> </tr> <tr> <td>Not Executed</td> <td>The test is associated to a requirement and is submitted for execution but not executed yet.</td> </tr> </table> | Passed | The test passed. | Failed | The test failed. | N/A | The test is associated to a requirement but is not submitted for execution. | Not Executed | The test is associated to a requirement and is submitted for execution but not executed yet. |
| Passed | The test passed. | | | | | | | | |
| Failed | The test failed. | | | | | | | | |
| N/A | The test is associated to a requirement but is not submitted for execution. | | | | | | | | |
| Not Executed | The test is associated to a requirement and is submitted for execution but not executed yet. | | | | | | | | |

Progress Reports

The following progress reports are available for the **Requirements** area:

| | |
|--|--|
| Requirements Coverage Across Builds 'X' and 'Y' | Represents a summary of all requirements by current requirement coverage. The returned group of requirements is restricted by the <code>Priority</code> parameter, which specifies the lowest requirement priority that is considered in the data. |
| Requirements Coverage Over the Past 'X' Days | Represents a trend in requirements coverage by considering overall requirements coverage over 'X' days. |
| Specific Requirements Coverage Over the Past 'X' Days | Represents a trend in requirements coverage by considering specific requirements coverage over 'X' days. |

Document Reports

The following document reports are available for requirements:

| | |
|--|--|
| All Requirements | All requirements are represented with full requirement information. |
| Requirement with Child Requirements | The selected requirement is shown with its requirement ID. Full details regarding the child requirements of the requirement are displayed. |

All Related Issues Report

The **All Related Issues** report provides a detailed list of all issues related to the assigned tests for a requirement, and explains the relationship between requirements, the assigned tests, and issues that have occurred. The input parameter for the **All Related Issues** report is the identifier of the requirement.

Coverage

This section describes how to cover your requirements with tests.

Coverage Modes

Silk Central provides two modes for coverage. Full coverage and direct coverage. The status of a requirement in the full coverage mode considers the status of the child requirements and offers a cumulative view of the tests that cover the requirement. If one or more child requirements have the status `Not Covered`, then the full coverage status of the selected requirement is also `Not Covered`, even if the coverage status of the requirement is `Covered`. The full coverage mode enables easy evaluation of whether or not requirements are covered by tests.

The status of a requirement in the direct coverage mode is calculated based only on the status of the tests that are directly assigned to the requirement. Child requirements are not considered in calculations.

Switching Between Coverage Modes

To switch between full- and direct-coverage modes:

1. In the menu, click **Requirements > Details View** .
2. Click  /  to respectively view direct coverage or full coverage.

Coverage Page

Requirements > Details View > Coverage

The **Coverage** page, which you can only access in **Details View**, displays basic properties of the selected requirement or project. It also displays the status of all tests that are assigned to the requirement, including

the number and percentage of Passed, Failed, Not Executed, and Not Covered tests. A summary of all assigned tests is listed under **Total**. Requirements that are not covered by tests are listed as Not Covered.

The **Details View** page displays the same coverage status information in a “heat field” chart.

To view the status of all tests that are assigned to child requirements of the selected requirement in addition to all tests that are directly assigned to the requirement, check the **Full coverage** check box.



Note: In both **Details View** and the **Coverage** page, test totals accumulate to the parent level. For example, requirement totals include tests from child requirements and project totals include tests from all requirements.

The **Coverage** page displays the following properties for each selected requirement or project:

| Property | Description |
|-----------------------------|---|
| Requirement Name | Name of the selected requirement or project. |
| Project Name | |
| Priority | Priority that has been assigned to the selected requirement. |
| Risk | Risk that has been assigned to the selected requirement. |
| # Requirements (Calc.) | Total number of all covered requirements. Not included in this number are uncovered requirements and folders that do not have a test assigned to them. For example folders that inherit the coverage of their child requirements but are not actually a requirement themselves. |
| # Requirements (Total) | Total number of all requirements beneath the selected entity, including folders. |
| Requirement Status | Status of the selected requirement or project. |
| Project Status | <ul style="list-style-type: none"> • Passed • Failed • Not Executed • Not Covered |
| # Requirements Passed | Total and percentage of requirements in the project that have tests that have passed. |
| # Requirements Failed | Total and percentage of requirements in the project that have tests that have failed. |
| # Requirements Not Executed | Total and percentage of requirements in the project that have tests that have not been executed. |
| # Requirements Not Covered | Total and percentage of requirements in the project that are not covered by tests. |
| Total | Total number of requirements in the project. |

Flags

Flags indicate that a requirement has changed or needs to be reviewed. Requirements with flags appear in the requirements tree with flag icons. There is also a section in the requirement properties to view flag information.

You can add and remove flags as needed and others will see the flag when they look at the requirement. You can also flag a requirement and all of its children. There can only be one flag on an item at a time and

setting a flag on an item that already has a flag will overwrite the flag. Changes to the requirement flag are maintained in the requirements change history.

If you are integrating with an external requirements management tool, you will also see flags from new or updated external requirements. The following are scenarios about how flags are applied for integrated requirements:

- If you add a requirement to the requirements management tool after initial synchronization, then a subsequent synchronization is performed, when the new requirement is added to Silk Central, it will display with an *added* flag to indicate it was created during an update.
- If you edit a requirement that has been synchronized to Silk Central, the requirement in Silk Central will be flagged with a *updated* flag to indicate it was updated.

Setting a Flag

Set a flag on a requirement to indicate to others that the requirement has changed or needs to be reviewed.

To set a flag on a requirement:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement in the **Requirements** tree.
3. Right click the requirement and choose **Flag > Set Flag** . The **Set Flag** dialog box opens.
4. Type a comment for the flag in the **Comment** field.
5. To set the flag for all children of the currently selected requirement, click the **Set flag on child requirements** check box.
6. Click **Ok**.



Note: The flag information is available on the **Properties** page of the requirement.

Clearing a Flag

Clear a flag from a requirement when all changes are acknowledged or the requirement is reviewed.

To clear a flag from a requirement:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement in the **Requirements** tree.
3. To clear a single flag: Right click the requirement and choose **Flag > Clear Flag** .
4. To clear the flag for all children of the selected requirement: Right click the requirement and choose **Flag > Clear Flag - Include Children** . The flag information is removed from the **Properties** page of the requirement.



Note: You can also clear the flag by clicking **Clear Flag** on the **Properties** page of the requirement.

Creating a Filter for Flagged Requirements

Create a filter for flagged requirements to see only the requirements in the **Requirements** tree that apply to the filter condition.

To create a filter for flagged requirements:

1. In the menu, click **Requirements > Details View** .
2. Click **New Filter**. The **New Filter** dialog box opens.
3. Select **Requirements Property** from the **Selection Criteria** list.
4. Select **Requirement State** from the **Property** list.
5. Select = from the **Operator** list.

6. Select one of the following values from the **Value** list:

| | |
|------------------------|--|
| None | No flag is selected |
| Flagged Manual | User manual added a flag to a requirement. |
| Flagged New | Requirement was created. |
| Flagged Updated | Requirement was updated. |
| Obsolete | Requirement is marked as obsolete. |

7. Click **Save** and **Apply**.



Note: If child requirements are flagged, but one or more parent requirements are not flagged, the full path to the flagged child requirements will display, but with the parent requirements disabled. You cannot set or clear the flag for parent requirements that are in this state.

Configuring Requirement Properties

Configure requirement properties, attachments, flags, and coverage.

Besides the given default properties, you can also create your own requirement properties in Silk Central. These custom properties can be edited in the **Edit Requirement** dialog box just like the default properties.

To create custom properties for the active project, click **Requirements > Requirement Properties** in the menu and click **New Requirement Property**.

Custom properties are displayed in the **Requirements Details View** on the **Properties** page and in the **Requirements Document View**.

For additional information on custom requirement properties, see *Custom Requirement Properties*.

Replacing Requirement Properties

To replace the property value of one or more requirements:

1. In the menu, click **Requirements > Details View**.
2. Click  (**Replace**) in the toolbar or right-click a requirement and click **Replace**. The **Replace** dialog box appears.
3. Select a requirement property from the **Find in** list.
This list contains all default and custom properties.
4. In the **Find what** area, define your search criteria.
The UI controls of the **Find what** area vary, based on the selected property. For example: If you select the property **Risk** two lists appear, if you select the property **Description** a text field and two check boxes appear.
5. In the **Replace with** area, enter the value that shall replace the identified data.
6. *Optional:* Check the **Case sensitive** check box to consider uppercase and lowercase letters.
7. *Optional:* Check the **Match whole word only** check box to include just complete standalone instances of the string.
8. Click **OK**. The first requirement that meets the search criteria is highlighted in the **Requirements** tree.
9. Click one of the following:
 - **Replace:** Replaces the property value of the highlighted requirement.
 - **Replace All:** Replaces the property value of all requirements that contain the defined value.
 - **Find Next:** Highlights the next requirement that contains the defined value.
 - **Find Previous:** Highlights the previous requirement that contains the defined value.

- **New Replace:** Opens the **Replace** dialog box again, where you can set new values.
- **Close:** Closes the dialog box.

 **Note:** If you click **Replace all**, all inherited properties will be overwritten. Child requirements also lose their inheritance setting in this case. Use **Replace** only on a parent requirement if you want the child requirements to inherit the new value.

Finding Requirement Properties

In the **Requirements** area, you can locate requirements that meet certain search criteria with the **Find** command. You can also replace identified property data with alternate data that you specify with the **Replace** command. Both commands offer **Find Next** and **Find Previous** functions that allow you to step through all identified properties.

To find a requirement:

1. In the menu, click **Requirements > Details View** .
2. Select  on the toolbar to open the **Find** dialog box.

 **Note:** This command can also be executed by right-clicking a requirement and selecting **Find**.

3. Select a requirement property from the **Find in** list.
This list contains all default and custom properties.
4. In the **Find what** area, define your search criteria.

The UI controls of the **Find what** area vary, based on the selected property. For example: If you select the property **Risk** two lists appear, if you select the property **Description** a text field and two check boxes appear.

5. Click **OK** to begin your search. The first requirement that meets the search criteria will be highlighted in the tree view. The first requirement that meets the search criteria will be highlighted in the tree view.
6. Click **Find Next** on the **Find** dialog box to advance to the next requirement in the list that meets your search criteria. Click **Find Previous** on the **Find** dialog box to return to the previous requirement in the list that meets your search criteria.

 **Note:** When you reach the last or first requirement that meets the search criteria and respectively click **Find Next** or **Find Previous**, the search starts from the beginning.

Requirement Properties Page

Requirements > Details View > Properties

The **Requirement Properties** page displays high-level information about the selected requirement or project.

When requirements are synchronized with an external requirements management system, items are sometimes marked with an exclamation mark (!). This means that the marked field is not mapped to the external requirements management system. Use the property mapping feature to map property fields. If you don't use the property mapping feature, only the name and the description of the requirements are mapped. For more information, see *Editing Property Mapping*.

Item values that are marked with an asterisk (*) are values that are inherited from the parent requirement.

Click **Edit Requirement Properties** to edit the displayed properties.

The following table only applies to requirements that have a flag set.

| Item | Description |
|------------|---|
| Flagged By | Indicates who set the flag and when the flag was set. |

| Item | Description |
|-------------------|--------------------------------------|
| Comment | Indicates the comments for the flag. |
| Clear Flag | Click to remove the flag. |

The following table lists the properties of the selected requirement or project that are displayed on the **Properties** page.

| Property | Description |
|-------------------|--|
| Requirement Name | Name of the requirement. |
| Requirement ID | Identifier of the requirement. |
| Description | Description of the requirement. |
| Priority | Priority that has been configured for the requirement. |
| Risk | Risk that has been configured for the requirement. |
| Reviewed | Current review status of the requirement. <i>Yes</i> or <i>No</i> . |
| Custom Properties | If custom properties have been configured for the requirement, they are listed here. |
| Document | Source document (if any) from which this requirement was derived. |
| Created On | Date on which this requirement was created. |
| Created By | Name of the user who created this requirement. |
| Changed On | Date on which this requirement was last updated. |
| Changed By | Name of the user who last updated this requirement. |

Requirement Attachments

You can upload multiple files or links as attachments to requirements or delete attachments. When you cut and paste requirements that have attachments, the attachments are automatically included with the copies.

The following attachment types are available:

- Uploaded Files
- References to UNC paths
- References to URLs, including StarTeam URLs

Attaching Files to Requirements

To attach a file to a requirement:

1. In the menu, click **Requirements > Details View**.
2. Select the requirement in the **Requirements** tree.
3. Click the **Attachments** tab.

When requirements management integration has been enabled between a Silk Central project and a Caliber project, the **Attachments** page includes an **Open Caliber** button, which enables you to manage requirement attachments directly in Caliber.

4. Click **Upload File**. The **Upload File** dialog box appears.
5. Click **Choose File** and select the file that you want to attach from your local file system.
6. Type a **Description** for the attachment.

7. Click **OK**. The attachment is uploaded to the server and associated with the selected requirement.

Attaching Links to Requirements

To attach a link to a requirement

1. In the menu, click **Requirements > Details View** .
2. Select the requirement in the **Requirements** tree.
3. Click the **Attachments** tab.

When requirements management integration has been enabled between a Silk Central project and a Caliber project, the **Attachments** page includes an **Open Caliber** button, which enables you to manage requirement attachments directly in Caliber.

4. Click **Attach Link**. The **Attach Link** dialog box appears.
5. Type the URL in the **Link** field.
6. Type a **Description** for the attached link.
7. Click **OK**. The link is associated with the selected requirement.

Viewing Requirement Attachments

To view a requirement attachment:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement in the **Requirements** tree.
3. Click the **Attachments** tab.

When requirements management integration has been enabled between a Silk Central project and a Caliber project, the **Attachments** page includes an **Open Caliber** button, which enables you to manage requirement attachments directly in Caliber.

A list of all attachments that are associated with the requirement opens. Each attachment name serves as a link. File-attachment links open **Save As** dialog boxes, enabling you to download attachments to your local file system. Link-attachments link directly to link destinations in newly spawned browser windows.

Downloading Files from Requirements

To download all files from a requirement:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement in the **Requirements** tree.
3. Click the **Attachments** tab.

When requirements management integration has been enabled between a Silk Central project and a Caliber project, the **Attachments** page includes an **Open Caliber** button, which enables you to manage requirement attachments directly in Caliber.

4. In the **Attachments** section, click **Download All Files**. All attached files are downloaded to your local disk as a zip archive, depending on how you configured your browser's download settings.



Note: To prevent from excessive downloads, **Download All Files** is disabled if **Include Child Attachments** is checked.

Deleting Attachments from Requirements

To delete a requirement attachment:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement in the **Requirements** tree.

3. Click the **Attachments** tab.

When requirements management integration has been enabled between a Silk Central project and a Caliber project, the **Attachments** page includes an **Open Caliber** button, which enables you to manage requirement attachments directly in Caliber.

4. Click  in the **Actions** column of the attachment that you want to delete.
5. Click **Yes** on the confirmation dialog to delete the attachment from the project.

 **Note:** Only one attachment at a time can be deleted.

Requirement Attachments Page

Requirements > Details View > Attachments

The **Attachments** page lists files and links that are attached to the selected requirement. The attachments are displayed in the order in which they are uploaded, though the list of attachments can be sorted by **Name**, **Created On**, and **Created By** properties.

 **Note:** To display any attachments that are associated with child requirements of the selected requirement, check the **Include Child Attachments** check box.

 **Note:** The file icons indicate whether documents are directly attached to the selected requirement, or whether they are attached to a child requirement of the selected requirement.

Single icon The file is directly attached to the selected requirement.

Double icon The file is attached to a child requirement of the selected requirement.

The **Attachments** page displays the following columns for each listed attachment:

| Column | Description |
|-----------------|---|
| Actions | Click  to delete the attachment. |
| Attachment Icon | Depends on the type of the attachment. |
| Name | Name of the attachment. |
| Size | Size of the attachment. |
| Description | Description of the attachment. |
| Created On | When the attachment was created. |
| Created By | User who created the attachment. |

Associating Tests with Requirements

Associate your requirements and tests to enable measuring how well your software covers the specified requirements.

Assigning Tests from Grid View to Requirements

To assign one or more tests from the test area **Grid View** to one or more requirements:

1. In the menu, click **Tests > Details View** .
2. Click  on the toolbar.
3. Select the tests that you want to assign to requirements.

You can use your keyboard's **Ctrl** and **Shift** keys to select multiple tests using standard browser multi-select functions.

4. Right-click the selected tests and select **Save Selection**.
5. In the menu, click **Requirements > Details View**.
6. Select the requirement to which you want to assign the selected tests.
7. Click the **Assigned Tests** tab.
8. Click **Assign Saved Selection**.



Note: Only tests that reside in the requirements test container are assigned. You can assign the selected tests to more than one requirement but you cannot assign them into requirements in a different project. The selection persists until you make a different selection or close Silk Central.

Assigning Tests to Requirements Manually

To manually assign tests to a requirement:

1. In the menu, click **Requirements > Details View**.
2. Select the requirement in the **Requirements** tree.
3. Click the **Assigned Tests** tab. All tests that are available for assignment are displayed in the **Test** window.
4. If you have created a test filter, you can select it from the filter list above the **Test** window.
To create a new test filter, navigate to the **Tests** area and click **New Filter** on the toolbar.
5. Select the tests that you want to assign.
You can use your keyboard's **Ctrl** and **Shift** keys to select multiple tests using standard browser multi-select functions.
6. Drag your selection and drop it into the **Assigned Tests** area, or click  left of any test.
Assigning a test container or test folder assigns all tests that are located in those containers or folders to the selected requirement. Tests that are located within sub-folders of those containers and folders are also assigned.



Tip: Right-click a node to expand and collapse it.

Locating Assigned Tests in the Tests Tree

To locate an assigned test in the **Tests** tree:

1. In the menu, click **Requirements > Details View**.
2. Click  in the toolbar.
3. Select a requirement in the **Requirements** tree that has at least one test assigned to it.
4. Click the **Assigned Tests** tab.
5. In the **Actions** column of a test, click  to find out in which test folder or container the test is stored in. The corresponding test folder or container is highlighted in the **Tests** window.

Removing Test Assignments

To remove a test assignment:

1. In the menu, click **Requirements > Details View**.
2. Click  in the toolbar.
3. Select a requirement in the **Requirements** tree that has at least one test assigned to it.
4. In the **Actions** column of the assigned test you want to remove, click , or use **Ctrl+Click** or **Shift+Click** to select multiple tests and press **Del**.



Note: To remove all test assignments from the selected requirement, click **Remove All**.

Generating Tests from Requirements Details View

You can generate tests directly out of the **Requirements** tree and assign tests to specific requirements. The **Requirements** tree serves as a template for the test folder/test structure of the new **Tests** tree.

To generate a new test from the **Details View**:

1. In the menu, click **Requirements > Details View**.
2. Right-click the requirement or project node that you want to convert into a test and select **Generate Tests**. The **Generate Tests from Requirements** dialog box appears. This dialog box enables you to specify whether the leaves, which means the lowest-level nodes, of the selected requirements sub-tree should be converted into tests or test folders and whether the tree should be generated into a new test container or an existing container.
3. Enter a name for the new test container in the **Enter Name** field and select a product from the **Select Product** list to create the container within the active Silk Central project.
The **Select Product** list is populated with the products that are configured by a project manager. For detailed information, see *Managing Products* or ask your project manager.
4. If you have defined a source control profile, select the source control profile you want to use for managing the test sources from the **Select Source Control Profile** list.
For detailed information on source control profiles, see *Source Control Profiles* or ask your Silk Central administrator.
5. To include all child requirements of the selected requirement in the test, check the **Include Child Requirements** check box.
The check box is checked by default.
6. To have the new tests automatically assigned to the requirements from which they are created, check the **Assign newly generated tests to Requirements** check box.
If this option is not selected, tests must be manually associated with requirements.



Note: This option is not available when checking **Generate test folders from requirement tree leaves**.

7. Click **OK** to create the test.
The new test has the same structure as the **Requirements** tree.
8. A message box displays. Click **Yes** to view the test in the **Tests** area, or click **No** to remain in the **Requirements** area.

Assigned Tests Page

Requirements > Details View > Assigned Tests

The **Assigned Tests** page lists all tests that are assigned to the selected requirement. If a test is assigned to more than one execution plan and the **Display in Execution Context** check box is checked, the test is displayed in a separate line of the grid for each execution plan.

All tests that are available for assignment are displayed in the right-hand **Tests** tree. To assign a test to the selected requirement, double-click on the entry in the tree or click  to the left of the entry.

If you have created a test filter, you can select it from the filter list above the **Tests** tree. To create a new test filter, navigate to the **Tests** area and click **New Filter** on the toolbar.

To automatically assign newly generated tests to the requirements from which they are generated, check the **Assign newly generated Tests to Requirements** check box on the **Generate Test from Requirements** dialog box.

The default view displays only those tests that are assigned directly to the selected requirement. Check the **Full Coverage** check box to additionally display all tests that are assigned to child requirements of the

selected requirement. When **Full Coverage** is checked, tests that are not assigned to the selected requirement, but are assigned to a child of the requirement, are displayed with an overlay over the test-type icon.

If you have copied a set of tests from **Tests > Grid View** to the clipboard, click **Assign Saved Selection** to assign the entire set to the selected requirement. Click **Remove All** to remove all assigned tests from the requirement.

To assign existing issues to one or more tests directly in the grid, right-click a test and click **Assign Existing Issues**. You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

The **Assigned Tests** page displays the following columns for each listed test:

| Column | Description |
|-----------------------|--|
| Actions | <p>You can perform the following actions on an assigned test:</p> <ul style="list-style-type: none"> Click  to remove the assigned test from the requirement. Click  to locate the assigned test in the right-hand Tests tree. |
| Test | Name of the assigned test. Click to view and edit the test. For configuration testing, the test is separately listed for each configuration. |
| Execution Plan | This column is only displayed if the Display in Execution Context check box is checked. Name of the execution plan that includes the assigned test. For configuration testing. |
| Execution Plan Parent | The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed. |
| Status | <p>Status of the assigned test in each execution plan to which the test is assigned. To display the status of all runs of the displayed tests in each execution plan, check the Display in Execution Context check box.</p> <ul style="list-style-type: none"> Passed Failed Not Executed N/A <p>For tests or test packages that are included in a configuration suite, the status is an aggregation of all statuses of all configurations within the suite:</p> <ul style="list-style-type: none"> If the test is assigned only to the configuration suite, the status is <i>Passed</i> only if it is passed in all configurations. If the status is not <i>Passed</i> in all configurations, the status of the worst run is propagated, with the following top-down priority: <ol style="list-style-type: none"> N/A Not Executed Failed Passed <p>The status of a requirement that is assigned to the test is <i>Failed</i> if the test fails in one or more of the configurations.</p> <ul style="list-style-type: none"> We do not recommend assigning the test to a configuration suite and one or more simple execution plans. However, in this case, the aggregate of the configuration statuses is one status, and the last statuses of the execution plans are other statuses. The status that was executed last determines the last status of the test. If the test is assigned to a simple execution plan only, the status is updated in response to the status of the last test run. |

| Column | Description |
|----------------|---|
| Last Execution | Time and date of the last execution of the test. |
| Issues | This column is only displayed if the Display in Execution Context check box is not checked. Issues that are associated with this test. |
| Path Info | The path to the test in the test folders structure. |
| Workflow State | Current workflow state of the test. For more information, see <i>Test Asset Workflow</i> . This column is hidden by default. |

Requirements Import

Silk Central enables you to import requirements from Microsoft Word and Excel. You can import a requirement into the **Requirements** area and then maintain all included requirements in Silk Central. The import is a one-time event.

Importing Requirements from a Word File

Before you can import requirements, you need to create a requirements file in Word. For more information, see *Word Requirements File*.

Silk Central uses the information stored in the Word file and in a mapping file to determine what data to store and captures only the information that is mapped with certain styles. Silk Central uses a mapping file to map the requirements in the Word file to the Silk Central requirements. A default mapping file named `RequirementsPropertiesDefaultMapping.xml` is preselected for the import. You can choose your own customized mapping file to map your requirements.

`docx` functionality: The functionality for `docx` is the same as for `doc` with the following improvements:

- Images and any other embedded object (OLE) (for example: PDF, Excel) are detected and imported as an attachment to a requirement.
- Descriptions may contain hyperlinks.

To import requirements from a Word file into the **Requirements** unit:

1. In the menu, click **Requirements > Details View**.
2. Right-click the requirement that you want to use as the parent node of the imported requirements and choose **Import from Word...**. The **Import Requirements from Word File** dialog box opens.
3. Select the mapping file from the **Choose Mapping File** list box. The default mapping file is `RequirementsPropertiesDefaultMapping.xml`.
4. *Optional:* Click  to edit the mapping file. Click  to delete the mapping file. For more information, see [Word Requirements File](#).



Note: These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.

5. Check the **One-Time Import** check box if this is a one-time bulk import of requirements, which will only be maintained in Silk Central after the import. Selecting this option imports the requirements directly under the selected node without creating a common root folder. You cannot use the **Update from Word...** option to update the imported requirements afterwards.
6. Either type the fully qualified path of the requirements file into the **Requirements File** field or click **Choose File** to search for the requirements file.
7. Click **OK**. A message informs you if the import was successful. Click **OK** to close the message.

Updating Requirements from a Word File

Silk Central enables you to update requirements that were imported from Microsoft Word. You can change the requirements in the Word document and then update them in the **Requirements** area. You can also change the mapping file to change the mapping of the requirements in the **Requirements** area.

 **Note:** You can only update a requirement for which a unique external ID is defined in the Word file that has previously been imported. Silk Central requires this ID to identify which requirement needs to be updated.

1. In the menu, click **Requirements > Details View**.
2. Right-click the parent node of the previously imported requirements (for example `Imported from file 'RequirementExample.docx'`) and choose **Update from Word...**. The **Update Requirements from Word File** dialog box opens.
3. Select the mapping file from the **Choose Mapping File** list box. The default mapping file is `RequirementsPropertiesDefaultMapping.xml`.
4. *Optional:* Click  to edit the mapping file. Click  to delete the mapping file. For more information, see [Word Requirements File](#).

 **Note:** These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.

5. Either type the fully qualified path of the requirements file into the **Requirements File** field or click **Browse** to search for the requirements file.
6. Click **OK**. A message informs you if the update was successful. Click **OK** to close the message.

Word Requirements File

Create a requirements file in Word to import requirements into Silk Central or to update them. In the requirements file, define requirements and their hierarchy, along with their priorities, risks, review status, and custom properties.

To map the requirements in your requirements file to the requirements in Silk Central, use an XML mapping file. The mapping file maps the styles in the Word document to the requirement properties in Silk Central. You can name the Word styles as you want, as long as you map them appropriately in your mapping file.

The requirements in the requirements file are separated by their name. For the mapping to work appropriately, the requirement name needs to be the first paragraph for any requirement in the file.

If you want to be able to update the requirements, you must define a style for the external ID and assign a unique external ID to each of the requirements in the requirement file. The following example shows a sample requirement, two subrequirements, and the corresponding external IDs:

| Word file | Applied styles |
|--|---|
| Customer Account Management - Basics | Formatted with Word style "Heading 1" |
| ID_1 | Formatted with Word style "ExternalID" |
| As user of the web shop I want to be able to ... | Formatted with Word style "Description" |
| Medium | Formatted with Word style "Priority" |
| High | Formatted with Word style "Risk" |
| No | Formatted with Word style "Reviewed" |
| Customer Account Management - Personal Details | Formatted with Word style "Heading 2" |
| ID_1.1 | Formatted with Word style "ExternalID" |
| As user of the web shop I want to be | "Description" |

| Word file | Applied styles |
|--------------------------------------|--|
| able to ... | Formatted with Word style "Priority" |
| Medium | Formatted with Word style "Risk" |
| Medium | Formatted with Word style "Reviewed" |
| No | |
| Customer Account Management - | Formatted with Word style "Heading 2" |
| Shipping Details | Formatted with Word style "ExternalID" |
| ID_1.2 | "Description" |
| As user of the web shop I want to be | Formatted with Word style "Priority" |
| able to ... | Formatted with Word style "Risk" |
| Medium | Formatted with Word style "Reviewed" |
| Medium | |
| No | |

Optionally, change the name of the mapping file in the **Save as** field. This file is saved on the application server in <Silk Central data directory>\OfficeImportMappingFiles\requirements \<client ID>, for example C:\ProgramData\SilkCentral\instance_<instance number>_<instance name>\OfficeImportMappingFiles.

Importing Requirements from an Excel File

Before you can import requirements, you need to create a requirements file in Excel. For more information, see *Excel Requirements and Mapping File*.

Silk Central uses the information stored in the Excel file and in a mapping file to determine what data to store and captures only the information that is mapped with certain column names. Silk Central uses a mapping file to map the requirements in the Excel file to the Silk Central requirements. A default mapping file named RequirementPropertiesXlsxDefaultMapping.xml is preselected for the import. You can choose your own customized mapping file to map your requirements.



Note: Importing attachments (images, graphs) is not supported.

To import requirements from an Excel file into the **Requirements** unit:

1. In the menu, click **Requirements > Details View**.
2. Right-click the requirement that you want to use as the parent node of the imported requirements and choose **Import from Excel...**. The **Import Requirements from Excel File** dialog box opens.
3. Select the mapping file from the **Choose Mapping File** list box. The default mapping file is RequirementPropertiesXlsxDefaultMapping.xml.
4. *Optional:* Click  to edit the mapping file. Click  to delete the mapping file. For more information, see [Excel Requirements and Mapping File](#).



Note: These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.

5. Check the **One-Time Import** check box if this is a one-time bulk import of requirements, which will only be maintained in Silk Central after the import. Selecting this option imports the requirements directly under the selected node without creating a common root folder. You cannot use the **Update from Excel...** option to update the imported requirements afterwards.
6. Either type the fully qualified path of the requirements file into the **Requirements File** field or click **Choose File** to search for the requirements file.
7. Click **OK**. A message informs you if the import was successful. Click **OK** to close the message.

Updating Requirements from an Excel File

Silk Central enables you to update requirements that were imported from Microsoft Excel. You can change the requirements in the Excel document and then update them in the **Requirements** area. You can also change the mapping file to change the mapping of the requirements in the **Requirements** area.

 **Note:** You can only update a requirement for which a unique external ID is defined in the Excel file that has previously been imported. Silk Central requires this ID to identify which requirement needs to be updated.

1. In the menu, click **Requirements > Details View**.
2. Right-click the parent node of the previously imported requirements (for example `Imported from file 'RequirementExample.xlsx'`) and choose **Update from Excel...** The **Update Requirements from Excel File** dialog box opens.
3. Select the mapping file from the **Choose Mapping File** list box. The default mapping file is `RequirementPropertiesXlsxDefaultMapping.xml`.
4. *Optional:* Click  to edit the mapping file. Click  to delete the mapping file. For more information, see [Excel Requirements and Mapping File](#).

 **Note:** These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.

5. Either type the fully qualified path of the requirements file into the **Requirements File** field or click **Browse** to search for the requirements file.
6. Click **OK**. A message informs you if the update was successful. Click **OK** to close the message.

Excel Requirements and Mapping File

Create a requirements file in Excel to import requirements into Silk Central or to update them. In the requirements file, define requirements and their hierarchy, along with their priorities, risks, review status, and custom properties. If the Excel file contains images, they are attached to the requirements during import. The position of the image anchor determines to which requirement the image is attached.

To map the requirements in your requirements file to the requirements in Silk Central, use an XML mapping file. The mapping file maps the columns in the Excel document to the requirement properties in Silk Central. You need the **Modify Office import mapping files** permission to edit the mapping file.

Property Mappings

Click **Edit Mapping** on the **Import Requirements from Excel File** or **Update Requirements from Excel File** dialog to map the requirements in your Excel file to Silk Central as follows:

- **Header Row:** Specify the row in your Excel requirements file which contains the column names to be used for mapping the properties.
- **First Data Row:** Specify the row that contains the first requirement. This row and all rows below are interpreted as requirements.
- **Separator Characters:** Specify the characters that should be interpreted as separators within property values. This is used for multi-list properties.
- **Name, Description:** Specify the column names in the header row that are mapped to requirement name and description, respectively.
- **External GUID:** Specify the column name in the header row that contains the unique ID within your requirements file for each requirement. These IDs can be used in **External Parent GUID** to create a hierarchy of your requirements.
- **External Parent GUID:** Specify the column name in the header row that contains the parent of a requirement. In your requirements file, enter the parent's **External GUID** to specify the parent of a

requirement. If this cell does not contain data, the requirement is added directly to the generated root requirement. The following rules apply:

- If a requirement has no parent ID set, it is added as a child to the generated root requirement
- If a requirement has an invalid parent ID, it is added as a child to the generated root requirement
- If a requirement has its own external ID set as parent ID, it is added as child to the generated root requirement
- If two or more requirements have a circular dependency through their parent IDs, they are not imported into Silk Central
- **Priority, Risk, Reviewed:** Specify the column names in the header row that are mapped to the localized values for these properties. Mapping the columns for these properties is mandatory.
- *Optional:* If you have defined custom properties, specify the column names in the header row for each of them to be mapped correctly.
- **Save as:** Specify the name of the file to be saved with your mapping information. This file is saved on the application server in <Silk Central data directory>\OfficeImportMappingFiles\requirements\xlsx\<client ID>, for example C:\ProgramData\SilkCentral\instance_<instance number>_<instance name>\OfficeImportMappingFiles\requirements\xlsx\1.

Sample Excel Requirements File

The following example shows a sample requirement, two subrequirements, and the corresponding external IDs. This sample Excel requirements file matches the default property mappings.

| 1 | RequirementName | RequirementDescription | ID | ParentID | Priority | Risk | Reviewed |
|---|--|--|--------|----------|----------|--------|----------|
| 2 | Customer Account Management - Basics | As user of the web shop I want to be able to ... | ID_1 | | Medium | High | No |
| 3 | Customer Account Management - Personal Details | As user of the web shop I want to be able to ... | ID_1.1 | ID_1 | Medium | Medium | No |
| 4 | Customer Account Management - Shipping Details | As user of the web shop I want to be able to ... | ID_1.2 | ID_1 | Medium | Medium | No |

Requirement Views

Silk Central offers several views and tools to manage your requirements.

Requirements Tree

Requirements are displayed, organized, and maintained through a hierarchical tree structure, the **Requirements** tree. Each node in the **Requirements** tree represents a requirement. Each requirement can have any number of child requirements associated with it. The **Requirements** tree enables you to organize requirements in any number of hierarchical levels.



Note: When the **Requirements** tree includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Page number links at the bottom of the tab allow you to browse through the elements included in the tree one page at a time. To display all elements as a single list, select the [All] link.

Keyboard Functions (Requirements)

The following actions can be performed on the selected nodes:

| Keys | Action |
|--------------------|--|
| Up | Move selection up |
| Down | Move selection down |
| Right | Expand element |
| Left | Collapse element |
| F2 | Edit element |
| Ctrl+C | Copy element |
| Ctrl+X | Cut element |
| Ctrl+V | Paste element |
| Del | Delete element |
| Ctrl+Up | Move element up |
| Ctrl+Down | Move element down |
| Click | Select an element and remember it as the current element. |
| Ctrl+Click | Toggle the selection status of the clicked element and add/remove additional elements with further clicks. |
| Shift+Click | Select the span from the currently-selected element to a newly selected element. |
| Shift+Up | Select element above. |
| Shift+Down | Select element below. |
| Enter | Assign element (only in assignment tree) |

Collapsing or Expanding the Requirements Tree

To collapse or expand the **Requirements** tree:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement folder in the **Requirements** tree and continue with one of the following options:
 - Click  to the left of the name of the requirement folder to expand the folder.
 - Click  to the left of the name of the requirement folder to collapse the folder.
 - Right-click on the folder and select **Expand** or **Collapse**. You can expand two, three, or all levels at once by using the command in the context menu

Filtering the Requirements Tree

If you filter a subtree, only the selected node, the parent nodes, and the child nodes are shown. All other nodes are temporarily hidden. This can be useful if you want to generate a PDF with a subset of all requirements.

To filter a subtree of the **Requirements** tree:

1. In the menu, click **Requirements > Details View** .
2. Right-click a requirement in the **Requirements** tree and click **Filter Subtree**.



Note: This automatically creates a custom filter in the filter list, which is placed in the toolbar. You can save this filter by clicking  (**Edit Filter**), clicking **Advanced**, and entering a name for the filter.

3. To disable the filter, select **<No Filter>** from the filter list in the toolbar.

Requirements Toolbar Functions

Requirements > Details View

Requirements > Document View

The requirements toolbar provides important commands for managing your requirements.

 **Note:** Some commands in the requirements toolbar are also available through context menus in the **Requirements** tree.

The following commands are included in the requirements toolbar:

| Command | Icon | Description |
|-----------------------|---|---|
| Details View |  | Displays the Details View , which enables you to drill deeply into the properties of a single requirement. |
| Document View |  | Displays the Document View , which shows selected properties of all requirements in a single view. |
| New Requirement |  | Adds a new requirement to the active project. |
| New Child Requirement |  | Adds a new child requirement to the selected requirement. |
| Edit |  | Edits the selected requirement. |
| Delete |  | Marks the selected requirement as obsolete or destroys it permanently. Obsolete requirements can be hidden, shown, and recovered. |
| Cut |  | Cuts a requirement from the Requirements tree and saves it to the clipboard. |
| Copy |  | Copys a requirement from the Requirements tree to the clipboard. |
| Paste |  | Pastes a requirement from the clipboard to the Requirements tree. |
| Paste as Child |  | Pastes a requirement from the clipboard as a child requirement to the currently selected requirement. |
| Move Up |  | Moves a requirement up within the Requirements tree. |
| Move Down |  | Moves a requirement down within the Requirements tree. |
| Find |  | Searches through all requirements in the active project based on configurable parameters. |
| Replace |  | Replaces instances of found values with a new value. |
| New Filter |  | Creates a new filter for the Requirements tree. |
| Filters | | Lists the available filters for the Requirements tree. |
| Edit Filter |  | Edits the selected filter. |
| Delete Filter |  | Deletes the selected filter. |
| Copy Filter |  | Copys the selected filter. |
| Show Changes |  | Shows recent changes to the requirements and their properties. |
| Acknowledge |  | Acknowledges changes to the requirements and their properties. |

| Command | Icon | Description |
|----------------------|---|--|
| Show Direct Coverage |  | Toggles between direct and full coverage modes. This button is disabled in the Details View . |
| Show Full Coverage |  | |
| Download as PDF |  | Generates and downloads a PDF with all currently visible nodes of the Requirements tree. This button is disabled in the Details View . |
| Download as XLSX |  | This function is enabled in the Document View . It generates an Excel sheet that includes all currently visible rows. |

Requirements Document View

Requirements > Document View

The **Document View** displays the status of all tests that are assigned to the active project, including the number and percentage of `Passed`, `Failed`, `Not Executed`, and `Not Covered` tests. The **Document View** displays this coverage status information in a “heat field” chart, with the following colors:

| Color | Description |
|--------|----------------------------------|
| Green | Passed tests. |
| Red | Failed tests. |
| Orange | Tests that are not executed yet. |
| Gray | Test with other status. |

Requirements that are not covered by tests are listed as `Not Covered`.

The **Document View** displays all custom requirement properties as columns.

 **Note:** Test totals accumulate to the parent level, for all currently displayed tests. If you apply a filter to the **Requirements** tree, the test totals are based on the filtered requirements only. For example, the requirement totals include tests from all child requirements that are selected with the filter, and the project totals include tests from all requirements that are selected with the filter.

The **Document View** individually saves the display settings for each user and each project.

Requirement History

Silk Central provides a complete history of all changes that are made to requirements. History information is read-only, and cannot be edited or permanently deleted.

The **Recent Changes** filter, which you can access by clicking  on the toolbar, enables you to efficiently view and acknowledge the latest changes and additions that have been made to requirements.

 **Note:** When you delete a requirement with the **Destroy permanently** option, you delete a requirement that is marked as obsolete, or if you acknowledge all recent changes, a change history entry is added to the history file of the project, to which the requirements belong.

Viewing Recent Changes

To view recent changes to requirements:

1. In the menu, click **Requirements > Details View**.
2. Click  to filter out all requirements except those that have been changed since your last change acknowledgement.



Note: The recent changes filter is selected automatically in the **Filter** list box.

3. When you have reviewed the changes, you can accept them by clicking . The acknowledge function resets the recent changes filter.
4. Click again to remove filtering and see all requirements.

Tracking the History of a Requirement

To view a requirement's history:

1. In the menu, click **Requirements > Details View**.
2. Select the requirement in the **Requirements** tree.
3. Click the **History** tab.
4. *Optional:* When requirements management integration between a Silk Central project and a Caliber project is enabled, you can click **Open Caliber** in the **History** page to view the history of synchronized requirements directly in Caliber.

The properties of all revisions that have been logged by Silk Central are displayed in tabular format.

Requirement History Page

Requirements > Details View > <Requirement> > History

The **History** page details the revision history of the selected requirement or project. The following actions generate requirement history entries:

- Adding a new requirement
- Editing a requirement
- Marking a requirement as obsolete
- Adding an attachment to a requirement
- Deleting an attachment from a requirement
- Importing or updating a requirement through MS Word or MS Excel

For each change, the page displays the following columns:

| Column | Description |
|-------------------|---|
| Rev. | Revision number. 1-n. |
| Changed On | Date and time of the change. |
| Changed By | User that made the change. |
| Notes | Automatically generated description of the change. For example, "deleted" or "created". |



Note: When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

Tests

This section explains how to manage tests in Silk Central.

A test, or test case, is a set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement [after IEEE 610]. Silk Central supports various test types, and tests can be structured in several ways. The **Tests** area enables you to maintain control over tests across the system development life-cycle. You can create, schedule, and manage both automated tests and manual tests. You can upload and associate files and links as attachments to test containers and tests. You can easily associate found issues with the tests that led to their discovery. You can also track the full change history of the test in the area.

Configuring Test Settings and Integrations

Before you can start building up your testing framework, you need to at least configure a [project](#) and a [product](#) and, if you plan to create automated tests, also a [source control profile](#). You may also need to configure certain other settings, like filters, properties, attributes, etc. You need SuperUser, administrator, or project manager privileges to perform these tasks.

Managing Products, Components, Versions, and Builds

Silk Central enables you to create and organize products, product components, versions of products, and build numbers of product versions. Product specifications can then be associated with tests, versions and builds can be associated with execution plans. You can even import products and components that were defined previously in Issue Manager.

- *Components* are discrete product elements that are tracked separately for testing purposes.
- *Versions* are product releases that are tracked separately for development and testing purposes.
- *Builds* are iterations of versions that are tracked separately for development and testing purposes.

Silk Central also enables you to set up platform designations for tests, for example operating systems. As with product, version, and build setup, platforms can be assigned names, descriptions, and active/inactive status.

Managing Products

Describes how to configure products in Silk Central.

Adding Products

To add a new product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click **New Product**. The **New Product** dialog box displays.
4. Type a name for the new product in the **Name** field.
5. Type a description for the product in the **Description** field.
6. Check the **Active** check box to make this product available for association with tests.
7. Click **OK** to save the product.



Note: When creating a product, Silk Central automatically creates a new default version 1.0 and a new default build 1 for the new product.

Editing Products

To edit an existing product:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. In the **Actions** column of the product that you want to edit, click . The **Edit Product** dialog box displays.
4. Make all required changes to the **Name**, **Description**, and **Active** status of the selected product, then click **OK**.

Activating and Deactivating Products

To activate or deactivate an existing product:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click the **Status** icon associated with the product you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
4. Confirm the message to toggle the product status to *active* or *inactive*.

Sorting the Products List

To move a product up or down in the list or sort the products list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. In the **Actions** column of the item you want to move, click or .
4. To sort the list alphabetically by name, click **Sort by Name**.

Deleting Products

To delete an existing product:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Deactivate the product you want to delete.
For additional information, see *Activating and Deactivating Products*.
4. In the **Actions** column of the product that you want to delete, click . A confirmation dialog box displays.
5. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Product Configuration Page

Administration > Products, Versions, and Builds > Products

Use this page to configure products, components, versions, and builds.

Click **New Product** to create a new product. Click **Sort by Name** to sort the products alphabetically by name.

For each listed product, the page displays the following columns:

| Column | Description |
|--------------------|--|
| Actions | <p>You can perform the following actions on a product:</p> <ul style="list-style-type: none">  Moves the product up one row in the list.  Moves the product down one row in the list.  Edit the name, description, and status of the product.  Deletes the product permanently. Products need to be inactive before you can delete them, and deletion is not allowed if a product is already associated to a test. |
| Name | The name of the product as it displays in the GUI and in reports. Click the name of a product to modify the product's components, versions, and builds. |
| Description | A textual description of the product. |
| Status | The status of the product, <i>Active</i> or <i>Inactive</i> . Click the status to toggle between <i>Active</i> and <i>Inactive</i> . |
| Created On | Date when the product was created. |
| Created By | The user who created the product. |

For a selected product, the page displays the details of the included components, versions, and builds.

The page displays the following columns for the components of the product:

| Column | Description |
|--------------------|---|
| Actions | You can delete the component in this column. |
| Name | The name of the component. Click to open the Edit Component dialog box. |
| Description | The description of the component. |
| Created On | Date when the component was created. |
| Created By | The user who created the component. |
| Status | The status of the component. <i>Active</i> or <i>Inactive</i> . Click to toggle the status. |

Click **New Component** to create a new component. Click **Sort by Name** to sort the components alphabetically by name. Click **Update** to update the components list.

The page displays the following items for the versions of the product:

| Item | Description |
|--------------------|---|
| Actions | You can delete the version in this column, or move the version up or down in the list. |
| Name | The name of the version. Click to open the Edit Version dialog box. |
| Description | The description of the version. |
| Created On | Date when the version was created. |
| Created By | The user who created the version. |
| Status | The status of the version. <i>Active</i> or <i>Inactive</i> . Click to toggle the status. |

Click **New Version** to create a new version. Click **Sort by Name** to sort the versions alphabetically by name.

The page displays the following items for the builds of the product:

| Item | Description |
|--------------------|---|
| Actions | You can delete the build or tag the build. |
| Name | The name of the build. Click to open the Edit Build dialog box. |
| Description | The description of the build. |
| Created On | Date when the build was created. |
| Created By | The user who created the build. |
| Status | The status of the build. <i>Active</i> or <i>Inactive</i> . Click to toggle the status. |

Click **New Build** to create a new build. Click **Sort by Name** to sort the builds list alphabetically by name. Click **Update** to update the components list.



Note: You have to select a specific version in the **Builds** list box to enable these buttons.

Click **Back** to return to the product list.

Managing Components

Components are discrete product elements that are tracked separately for testing purposes.

Adding Components

To add a new component to a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click **New Component**. The **New Component** dialog box displays.
5. Type a name for the new component in the **Name** field.
6. Type a description for the component in the **Description** field.
7. Select the **Type** of the component.
Available values are *Software* and *Documentation*.
8. Check the **Active** check box to make this component available for association with products.
9. Click **OK** to save the component, or click **Cancel** to abort the operation.

Editing Components

To edit an existing component within a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Select the component you want to edit.
5. The **Edit Component** dialog box displays, detailing the **Name**, **Description**, and **Active** status of the selected component. Make all required changes, then click **OK**.



Note: The type of a component can only be edited if you deactivate the component first. For additional information, see *Activating and Deactivating Components*.

Activating and Deactivating Components

To activate or deactivate an existing component:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click the **Status** icon associated with the component you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
5. Confirm the message to toggle the component status to *active* or *inactive*.

Sorting the Components List

To move a component up or down in the list or sort the components list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Component Type** list box, select the component type for which you want to see the components.
5. In the **Actions** column of the item you want to move, click  or .
6. To sort the list alphabetically by name, click **Sort by Name**.

Deleting Components

To delete an existing component from a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Deactivate the component you want to delete.
For additional information, see *Activating and Deactivating Components*.
5. In the actions column of the component, click  . A confirmation dialog box displays.
6. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Managing Versions

Versions are product releases that are tracked separately for development and testing purposes.

Adding Versions

To add a new version to a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click **New Version**. The **New Version** dialog box displays.
5. Type a name for the new version in the **Name** field.
6. Type a description for the version in the **Description** field.
7. Check the **Active** check box to make this version available for association with products.
8. Click **OK** to save the version, or click **Cancel** to abort the operation.

Editing Versions

To edit an existing version within a product:

1. In the menu, click **Administration > Products, Versions and Builds** .

2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Select the version you want to edit. The **Edit Version** dialog box displays.
5. Make all required changes to the **Name**, **Description**, and **Active** status of the selected version, then click **OK**.

Activating and Deactivating Products

To activate or deactivate an existing product:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click the **Status** icon associated with the product you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
4. Confirm the message to toggle the product status to *active* or *inactive*.

Sorting the Versions List

To move a version up or down in the list or sort the versions list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Actions** column of the item you want to move, click  or .
5. To sort the list alphabetically by name, click **Sort by Name**.

Deleting Versions

To delete an existing version from a product:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Deactivate the version you want to delete.
For additional information, see *Activating and Deactivating Versions*.
5. In the **Actions** column of the product that you want to delete, click . A confirmation dialog box displays.
6. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Managing Builds

Builds are iterations of versions that are tracked separately for development and testing purposes.

Adding Builds

To add a new build to a version:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Builds for Version** list box, select the version for which you want to add a build.
5. Click **New Build**. The **New Build** dialog box displays.

6. Type a name for the new build in the **Name** field.
7. Type a description for the build in the **Description** field.
8. Check the **Active** check box to make this build available for association with versions.
9. Click **OK** to save the build, or click **Cancel** to abort the operation.

Editing Builds

To edit an existing build within a version:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Select the build you want to edit.
5. The **Edit Build** dialog box displays, detailing the **Name**, **Description**, and **Active** status of the selected build. Make all required changes, then click **OK**.
6. Check the **Tagged** check box to tag the build.

Activating and Deactivating Builds

To activate or deactivate an existing build:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click the **Status** icon associated with the build you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
5. Confirm the message to toggle the build status to *active* or *inactive*.

Sorting the Builds List

To move a build up or down in the list or sort the builds list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Builds for Version** list box, select the version for which you want to see the builds.
5. In the **Actions** column of the item you want to move, click  or .
6. To sort the list alphabetically by name, click **Sort by Name**.

Tagging Builds

You can delete runs of tagged builds. If the selection of the runs that you are trying to delete contains tagged builds, then you get a warning and you have to confirm the action. Tagged builds will not get deleted if you perform a bulk delete (all or for a specific time span) by right clicking on the execution node.

To tag an existing build:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Builds** section, click  in the **Actions** column of the build that you want to tag. A confirmation dialog box displays.
5. Click **Yes** to continue with the tagging or click **No** to abort the tagging.

Deleting Builds

To delete an existing build from a version:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Deactivate the build you want to delete.
For additional information, see *Activating and Deactivating Builds*.
5. In the actions column of the build, click **X** . A confirmation dialog box displays.
6. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Managing Platforms

Set up platform designations for tests, for example operating systems. As with product, version, and build setup, platforms can be assigned names, descriptions, and active or inactive status.

Adding Platforms

To add a new platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. Click **New Platform**. The **New Platform** dialog box displays.
4. Type a name for the new platform in the **Name** field.
5. Type a description for the platform in the **Description** field.
6. Check the **Active** check box to make this platform available for association with tests.
7. Click **OK** to save the platform.

Editing Platforms

To edit an existing platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. In the **Actions** column of the platform that you want to edit, click  .
4. The **Edit Platform** dialog box displays, detailing the **Name**, **Description**, and **Active** status of the selected platform. Make all required changes, then click **OK**.

Activating and Deactivating Platforms



Note: Platforms that are associated with a test cannot be deactivated.

To activate or deactivate an existing platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. Click the **Status** icon associated with the platform you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
4. Confirm the message to toggle the platform status to *active* or *inactive*.

Sorting the Platforms List

To move a platform up or down in the list or sort the platforms list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. In the **Actions** column of the item you want to move, click  or .
4. To sort the platforms list alphabetically by name, click **Sort by Name**.

Deleting Platforms

To delete an existing platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. Deactivate the platform you want to delete.
For additional information, see *Activating and Deactivating Platforms*.
4. In the **Actions** column of the platform that you want to delete, click  . A confirmation dialog box displays.
5. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Platform Configuration Page

Administration > Platforms

Use this page to configure platforms.

To create a new platform, click **New Platform**. For additional information, see *Adding Platforms*.

To sort the platforms list alphabetically by name, click **Sort by Name**. For additional information, see *Sorting the Platforms List*.

For each listed platform, the page displays the following columns:

| Columns | Description |
|--------------------|---|
| Actions | <p>The user can perform the following actions on a platform:</p> <ul style="list-style-type: none">  Moves the platform up one row in the list.  Moves the platform down one row in the list.  Deletes the platform permanently. Platforms need to be inactive before you can delete them, and deletion is not allowed if a platform is already associated to a test. |
| Name | The name of the platform as it displays in the GUI and in reports. Click the name of a platform to modify the name, description, and status of the platform. |
| Description | A textual description of the platform. |
| Status | The status of the platform, <i>Active</i> or <i>Inactive</i> . Click the status to toggle between <i>Active</i> and <i>Inactive</i> . |
| Created On | Date when the platform was created. |
| Created By | The user who created the platform. |

Source Control Profiles

Source control profiles enable Silk Central to integrate with external source control systems. Defining source control profiles allows you to define where Silk Central's execution servers should retrieve program sources for test execution.

You need double the amount of free disk space on the execution server to accommodate the source files because Silk Central checks out the source control tree and then generates a working directory with the executable source files.

Additional source control systems can be configured by installing a custom plug-in. For detailed information, refer to the *Silk Central API Help*. The following source control systems are currently supported by Silk Central “out of the box”:

AccuRev

This section describes how to configure AccuRev source control profiles.

AccuRev is a software configuration management tool which addresses complex parallel and distributed development environments with its stream-based architecture, to accelerate your development process and improve asset reuse.

AccuRev versions that are currently supported for integration with Silk Central: AccuRev 7.2

Adding AccuRev Source Control Profiles

To create a AccuRev source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **AccuRev** from the **Source control system** list box.
6. Type a valid AccuRev **Username** and **Password**, if required.
7. Type the **AccuRev server** name of the AccuRev server you want to access. If you are not using AccuRev's default port, type `<server name>:<port number>`.
8. Type the name of the AccuRev **Stream**, for example `backing stream`.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing AccuRev Source Control Profiles

To modify a AccuRev source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
 2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
 3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
 4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
 5. Edit the **Username** and **Password**.
These credentials are used to access your repository.
 6. Edit the **AccuRev server** name of the AccuRev server you want to access. If you are not using AccuRev's default port, type `<server name>:<port number>`.
 7. Edit the name of the AccuRev **Stream**, for example `backing stream`.
 8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.
The working folder must be a local path. For example, `C:\TempSources\`.
-  **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.
9. Edit the **Project path** you want this profile to use.
 10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Apache Commons Virtual File System

This section describes how to configure Virtual File System (VFS) source control profiles.

A VFS is an abstraction layer on top of a more concrete file system. The purpose of a VFS is to allow client applications to access different types of concrete file systems in a uniform way. Apache Commons VFS provides a single API for accessing various file systems. It presents a uniform view of the files from various sources. The protocols that are currently supported for VFS by Silk Central are:

Protocol Description

- | | |
|-------------|---|
| http | Copies the given file. This protocol type is also supported for copying and unpacking ZIP, JAR, or other zipped files. It is required to specify a .zip file on a http server. For example, <code>zip:http://myTestServer/myTests.zip</code> . The .zip file will be extracted on the execution server. |
| ftp | Copies the given file. This protocol type is also supported for copying and unpacking ZIP, JAR, or other zipped files. |
| smb | Server Message Block (smb) copies all files and folders. This protocol can be used instead of a UNC profile. For example, the VFS smb path <code>smb://server-name/shared-resource-path</code> is equivalent to the UNC path <code>\\server-name\shared-resource-path</code> . |

 **Note:** When you create a new ProcessExecutor test that uses VFS for source control, you need to specify the complete path to the executable in the **Executable Name** field.

Adding VFS Source Control Profiles

To create a VFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **VFS** from the **Source control system** list box.
6. Type the URL of the VFS **Repository** you want to access. This field is case sensitive.
Specify the appropriate protocol type in the URL:

| | |
|-------------|--------------------------|
| FTP | ftp://<FTP server URL> |
| HTTP | http://<HTTP server URL> |
| SMB | smb://<Samba server URL> |

 **Note:** HTTP, FTP and SMB are also supported for zipped files. In order to point to a zipped file the URL must be adjusted to <zipped file type>:<protocol>://<server URL pointing to zipped file> to include the type of the zipped file. For example, zip:http://193.80.200.135/<path>/archive.zip or jar:http://193.80.200.135/<path>/archive.jar.

7. Type a valid VFS **Username** and **Password**.
These credentials will be used to access your VFS repository. The SMB protocol allows including the domain name in the username in the following form: domain/username.
8. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.
The working folder must be a local path. For example, C:\TempSources\.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

9. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing VFS Source Control Profiles

To modify a VFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.

4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Edit the URL of the VFS **Repository** you want to access.



Note: This field is case sensitive.

6. Edit the **Username** and **Password**.

These credentials are used to access your repository.

7. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

8. Edit the **Project path** you want this profile to use.

9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Git

This section describes how to configure Git source control profiles.

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed. Every Git working directory is a full-fledged repository with complete history and full version tracking capabilities, not dependent on network access or a central server.

Adding Git Source Control Profiles

To create a Git source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.

4. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.

5. Select **Git** from the **Source control system** list box.
6. Type the URL of the Git **Repository** you want to access.



Note: This field is case sensitive.

7. Type a valid Git **Username** and **Password**, if required.
8. *Optional:* Specify a **Branch** to check out a specific branch. Leave this field empty to check out the master branch.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.



Tip: If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan. See *Creating Execution Plans* for detailed information. This will overwrite the specified **Branch** in the Git source control profile.

Editing Git Source Control Profiles

To modify a Git source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Type the URL of the Git **Repository** you want to access.



Note: This field is case sensitive.

6. Edit the **Username** and **Password**.
These credentials are used to access your repository.
7. *Optional:* Specify a **Branch** to check out a specific branch. Leave this field empty to check out the master branch.
8. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.



Tip: If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan. See *Creating Execution Plans* for detailed information. This will overwrite the specified **Branch** in the Git source control profile.

Microsoft Team Foundation Server

This section describes how to configure Team Foundation Server (TFS) source control profiles.

TFS versions that are currently supported for integration with Silk Central: Team Foundation Server 2015, 2017



Note: To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).

Adding TFS Source Control Profiles

To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).



Note: If your TFS project has a Git repository configured, you have to create a Git source control profile instead of a TFS source control profile. Use `.` as the user name and a token that you have generated in TFS as the password for the Git profile. For additional information on creating a Git source control profile, see [Adding Git Source Control Profiles](#).

To create a TFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **Team Foundation Server** from the **Source control system** list box.
6. Enter the **URL** of your TFS installation.
For example, `http://tfsserver:8080/tfs`.
7. Type the **Domain** of the TFS user.
8. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.
The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing TFS Source Control Profiles

To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).

To modify a TFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the URL of the TFS **Repository** you want to access.
6. Edit the **Username** and **Password**.
These credentials are used to access your repository.
7. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

8. Edit the **Project path** you want this profile to use.
9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Silk Test Workbench

This section describes how to configure Silk Test Workbench source control profiles.

Silk Test Workbench is an automated testing tool that accelerates the functional testing of complex applications. Silk Test Workbench provides support for testing applications developed in a wide variety of development tools including Java, .NET, browser-based web applications, and COM components, including both ActiveX controls and automation objects. With Silk Test Workbench, you can record user sessions

with your applications to create tests, enhance the test by adding validation and test logic, and play back tests to ensure that the applications work as expected.



Note: To add a Silk Test Workbench source control profile, the Silk Test Workbench client needs to be installed on every execution server on which you want to use the source control profile.

Adding Silk Test Workbench Source Control Profiles

To create a Silk Test Workbench source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **Silk Test Workbench Test** from the **Source control system** list box.
6. Select a database type from the **Database type** list box, either SQL Server or Oracle.



Note: The Microsoft Office Access database type is not supported. When connecting to a Silk Test Workbench Oracle database, there is a Silk Test Workbench requirement that the TNS name must be the same as the host server name.

7. Enter the name of the Silk Test Workbench database server in the **Database server** field.
8. Enter the name of the Silk Test Workbench database in the **Database name** field.
9. Enter the port number of the Silk Test Workbench database server in the **Database port** field.
10. Enter the Silk Test Workbench database schema or owner name in the **Database schema/owner name** field.
11. Enter the name of the database user in the **Database user** field.
12. Enter the password for the database user in the **Database password** field.
13. *Optional:* Enter the name and password of a valid Silk Test Workbench user in the **Silk Test Workbench User name** and **Silk Test Workbench password** fields. If these fields are left empty, the Windows user that runs the execution server service is used.
14. Click **Retrieve**. All projects are listed in the **Projects** list.
15. Select one or more projects.
16. Click **OK**.

Editing Silk Test Workbench Source Control Profiles

To modify a Silk Test Workbench source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the Silk Test Workbench **Database type**.



Note: The Microsoft Office Access database type is not supported.

6. Edit the other settings as required.
7. Click **OK**.

StarTeam

This section describes how to configure StarTeam source control profiles.

StarTeam promotes team communication and collaboration through centralized control of all project assets. Protected yet flexible access ensures that team members can work whenever and wherever they like through an extensive choice of Web, desktop, IDE, and command-line clients. StarTeam offers a uniquely comprehensive solution that includes integrated requirements management, change management, defect tracking, file versioning, threaded discussions, and project and task management.

StarTeam versions that are currently supported for integration with Silk Central: StarTeam 15.x, 16.x, 17.0

Adding StarTeam Source Control Profiles

To create a StarTeam source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **StarTeam** from the **Source control system** list box.
6. Type the **Hostname** of your StarTeam server.
7. Type the **Port** that is to be used to connect to the StarTeam server.
If the port is not changed, use the default port 49201.
8. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
9. Specify if the profile supports **Encryption**.
10. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

11. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing StarTeam Source Control Profiles

To modify a StarTeam source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.

3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
 4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
 5. Choose from the following options:
 - Edit the **Hostname** of your StarTeam server.
 - Edit the port that is to be used to connect to the StarTeam server. If the port is not changed, use the default port 49201.
 - Specify if the profile supports **Encryption**.
 6. Edit the **Project path** you want this profile to use.
 7. Edit the **Username** and **Password**.
These credentials are used to access your repository.
 8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.
The working folder must be a local path. For example, C:\TempSources\.
-  **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.
9. Click **OK**.
Silk Central attempts a trial connection to the external system using the information you have provided.
-  **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Subversion

This section describes how to configure Subversion (SVN) source control profiles.

If you set a Source Control Label on the execution plan, you can check out specific Subversion branches or tags. Branches and tags are encoded in the Subversion URL. For example, `http://MyHost/svn/MyApp/trunk` is a Subversion URL and `http://MyHost/svn/MyApp/tags/build1012` is a Subversion tag.

To check out branches and tags in Silk Central, the main repository URL that is specified in the source control profile must contain the `trunk` directory. The `trunk` directory in the Subversion URL is then replaced with the label that is specified for the execution plan.

Adding Subversion Source Control Profiles

To create a Subversion source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Choose **Subversion** from the **Source control system** list box.
6. Type the URL of the Subversion **Repository** you want to access.
If you do not know the URL of the repository, consult your Subversion administrator.

 **Note:** To use SSH, add `ssh` to the URL, for example `svn+ssh://<hostname>:<port>`.

7. Type a valid **Username** and **Password**.

These credentials will be used to access your repository.

8. Type a valid **SSH username** and **SSH password** or **SSH keyfile**.

These credentials are used to access the SSH server. The password overrides the keyfile, so if you only have a keyfile, leave the **SSH password** field empty. If you use a keyfile, the path to the keyfile must be valid on every execution server that uses the source control profile.

9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing Subversion Source Control Profiles

To modify a Subversion source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.

2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.

3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.

4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Edit the **URL** of the Subversion repository you want to access.

If you do not know the URL of the repository, please consult your Subversion administrator.

 **Note:** If you cannot check out files, when editing the URL of the Subversion **Repository**, delete the source control mirrors directory on your execution server. For example, `C:\ProgramData\SilkCentral\SrcCtrlMirrors`.

 **Note:** To use SSH, add `ssh` to the URL, for example `svn+ssh://<hostname>:<port>`.

6. Edit the **Username** and **Password**.

These credentials are used to access your repository.

7. Type a valid **SSH username** and **SSH password** or **SSH keyfile**.

These credentials are used to access the SSH server. The password overrides the keyfile, so if you only have a keyfile, leave the **SSH password** field empty. If you use a keyfile, the path to the keyfile must be valid on every execution server that uses the source control profile.

8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

9. Edit the **Project path** you want this profile to use.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Silk TestPartner

This section describes how to configure TestPartner source control profiles.

TestPartner is a Micro Focus product that tests Web- and Microsoft Windows-based applications that use Microsoft technologies. TestPartner records user actions to quickly produce powerful tests. Each recorded test displays as a series of actions in clear, concise steps that can be easily understood by all testers, from novice to expert. You can record user sessions with the application, add validation functions, and replay the sessions later to ensure that the application works as expected.



Note: To add a TestPartner source control profile, the TestPartner client needs to be installed on every execution server on which you want to use the source control profile.

Adding TestPartner Source Control Profiles

To create a TestPartner source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **TestPartner** from the **Source control system** list box.



Note: When connecting to a TestPartner Oracle database, there is a TestPartner requirement that the TNS name must be the same as the host server name.

6. Select a database type from the **Database type** list box, either SQL Server or Oracle.
7. Type the name of the database server in the **Database server** field.
8. Type the name of the database in the **Database name** field.
9. Type the port number of the database server in the **Database port** field.
10. Type the database schema or owner name in the **Database schema/owner name** field.
11. Type the name of the database user in the **Database user** field.
12. Type the password for the database user in the **Database password** field.
13. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
14. Use the **Project path** field to filter which scripts from the database are available to use as tests. Click **Browse** to display the **Select Project Path** dialog box.

The **Select Project Path** dialog box provides a tree of three levels that you can choose from:

- First level** The entire TestPartner database. Note that if you select this level, the **Project path** field will remain blank.
- Second level** The TestPartner project.
- Third level** The script type within a specified TestPartner project.

Select one of the preceding options and click **OK**. The path will be added to the **Project Path** field.

15. Click **OK**.

Editing TestPartner Source Control Profiles

To modify a TestPartner source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Choose from the following options:
 - Edit the TestPartner **Database type**.
 - Edit the TestPartner **Database server**.
 - Edit the TestPartner **Database name**.
 - Edit the TestPartner number of the TestPartner database server.
 - Edit the TestPartner **Database schema/owner name**.
 - Edit the TestPartner **Database user**.
 - Edit the TestPartner **Database password**.
6. Edit the **Username** and **Password**.
These credentials are used to access your repository.
7. Edit the **Project path** you want this profile to use.
8. Click **OK**.

Universal Naming Convention

This section describes how to configure Universal Naming Convention (UNC) source control profiles.

Short for Universal Naming Convention or Uniform Naming Convention, UNC is a PC format for specifying the location of resources on a local area network (LAN). UNC uses the following format: `\\server-name\shared-resource-pathname`.

For example, to access the file `test.txt` in the directory `examples` on the shared server `silo`, you would write: `\\silo\examples\test.txt`.

You can also use UNC to identify shared peripheral devices, such as printers. The idea behind UNC is to provide a format so that each shared resource can be identified with a unique address.

UNC is only supported on Microsoft Windows operating systems. If you plan to use a non-Windows execution server you can use the Apache Commons VFS source control profile instead.

Adding UNC Source Control Profiles

To create a UNC source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **UNC** from the **Source control system** list box.
6. Type the **UNC path** that you want to access.



Tip: To avoid checking out unnecessary files, specify the folder at a hierarchy level where the sources reside that are actually required for test execution.

7. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`. If you do require a static working folder, make sure that it is not located within the specified **UNC path**, as this will lead to recursive checkouts.

8. Type a valid **UNC username** and **UNC password**.
These credentials are required to access the UNC path of the configuration file.
9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing UNC Source Control Profiles

To modify a UNC source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Type the **UNC path** that you want to access.

This is the path to the location where your test sources are located.



Tip: To avoid checking out unnecessary files, specify the folder at a hierarchy level where the sources reside that are actually required for test execution.

6. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running

parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir. If you do require a static working folder, make sure that it is not located within the specified **UNC path**, as this will lead to recursive checkouts.

7. Edit the **UNC username** and **UNC password**.

These credentials are required to access your configuration file's UNC path.

8. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Deleting Source Control Profiles

To remove a source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. In the **Actions** column of the issue tracking profile you wish to delete, click **Delete source control profile** A confirmation dialog box displays.
4. Click **Yes**. You are returned to the **Source Control** page.

Source Control Profiles Page

Project:<Project Name> > Project Settings > Source Control

Use this page to configure profiles to integrate external source control systems with Silk Central. Click **New Profile** to create a new source control profile. For each source control profile, the page displays the following columns:

| Item | Description |
|----------------|--|
| Name | The name of the profile as it displays in the Silk Central GUI and in reports. Click the name to edit a profile. |
| Type | The external source control system. |
| Working Folder | Local or mapped working folder to which temporary sources are checked out to. |
| Created On | Date when the source control profile was created. |
| Created By | The user who created the source control profile. |
| Changed On | Date when the source control profile was last modified. |
| Changed By | The user who last modified the source control profile. |
| Actions | Delete source control profile |

Step Properties

Use step properties to extend a manual test step with additional properties. Add a custom step property to a project in **Projects:<Project Name> > Project Settings** to add the property to all manual test steps in the project. Add a custom step property to a library in the **Libraries** tree to add a custom step property to all manual test steps in the library. Step properties can subsequently be edited alongside the default properties on the **Edit Manual Test Step** dialog box.

Creating Step Properties

To create a new step property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Step Properties** tab.
3. Click **New Property**. The **New Step Property** dialog box opens.
4. Enter a name for the new property in the **Name** field.



Note: Step property fields are always declared as type string.

5. Click **OK** to make your property available to all manual test steps in the selected Silk Central project.



Note: To create a step property for a library, select the library node in the **Libraries** tree and continue with the second step.

Editing Step Properties

To edit a previously created step property

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Step Properties** tab.
3. Click on the name of the property that you want to edit. The **Edit Step Property** dialog opens.
4. Edit the name of the property in the **Name** field.
5. Click **OK** to save your changes, or click **Cancel** to abort the operation without saving.

Deleting Step Properties

To delete a previously created step property

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Step Properties** tab.
3. Click **Delete Property** in the **Actions** column of the property that you want to delete. A confirmation dialog box displays, asking you to confirm the deletion.
4. Click **Yes** to complete the operation, or **No** to abort.

Step Properties Page

Project:<Project Name> > Project Settings > Step Properties

The **Step Properties** page lists all properties that can be populated into manual test steps across the active project. For each custom step property, the **Step Properties** page displays the following columns:

| Column | Description |
|---------|--|
| Name | Name of the step property. |
| Actions | The following actions can be performed on a property: <ul style="list-style-type: none">• Delete Property• Move Up• Move Down |

Attributes

You can use attributes to customize the information for tests and execution plans. While some attributes are made available by the integrated functionality of Silk Central, such as priority, components, and platforms,

you may want to define custom attributes to categorize tests and execution plans to your needs, or to make tests compatible with specific test cases.

Creating Attributes

To create an attribute:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Attributes** tab to view the list of current attributes.
3. Click **New Attribute**. The **New Attribute** dialog box displays.
4. Type a **Name** for the new attribute. This name will be displayed in list boxes when the attribute becomes available for use.
5. Type a **Description** for the new attribute.
6. Select the attribute **Type**. See the topic *Types for Attributes and Custom Requirements Properties* for descriptions of each type of attribute.
7. Optionally check the **Required (Tests)** check box. Required attributes must be set when creating a new test. Required attributes that are not yet set display with an orange background and a red border on the **New Test** dialog.

 **Note:** If the parent test has a certain attribute already set, the value is automatically filled in for the child test. However, the **Inherit from parent** check box is always unchecked by default. If the parent test does not require a certain attribute, the **Inherit from parent** check box is disabled.

 **Note:** If an attribute is set to **Inactive**, it is not required when creating a new test.

8. Optionally check the **Required (Execution Planning)** check box. Required attributes must be set when creating a new execution plan, configuration suite, or testing cycle. Required attributes that are not yet set display with an orange background and a red border on the creation dialog.

 **Note:** If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.

 **Note:** If an attribute is set to **Inactive**, it is not required when creating a new element.

9. Click **OK**.

Types for Attributes and Custom Requirements Properties

Silk Central supports the following types of attributes and custom requirement properties:

Date The **Date** type supports date values.

List The **List** type supports single select or multi-select lists. It has the following controls:

- **New Item:** Allows you to create entries in the list. Click to show the **New Item** dialog box. Enter the **Name** and **Numeric Weight** for the list item and click **OK**.

 **Note:** The value of the **Numeric Weight** field is used during filtering when the filter criteria uses any of the following operators: **>**, **>=**, **<**, or **<=**. When the filter criteria uses the **=** or **not** operators, the value from the **Name** field is used for comparisons.

- **Order:** Use the fields in this column to sort your attributes.
- **Allow multi-select:** Click this check box to allow users to select more than one values from the list.

 **Note:** You can update an existing single-select list to be a multi-select list. A multi-select list cannot be changed to being a single-select list though.

- **Actions:** Use the icons in this row to delete or to edit each item in the list.

Global List

The **Global List** type allows you to reuse items from a centrally managed **List of Values**. A global list lets you maintain a set of items in a single place and use them in multiple projects. These items can be managed (added, removed, or modified) on the **List Of Values Configuration Page**. All changes on the **List Of Values Configuration Page** affect all related **Global List** attributes and custom properties in all projects.

- **Allow multi-select:** Check this check box to allow users to select more than one item from the list.



Note: You can turn a single-select list into a multi-select list. But you can not turn a multi-select list into a single-select list.



Note: When you disable a value from a **List Of Values**, the corresponding **Global List** item remains where it is already used but does not appear any longer in the choice of items in **Edit** operations.

A List of Values can be grouped by another List of Values (the grouping is optional). In such a case, the attributes or custom properties are connected to each other. They are added, edited, and removed coherently. This means ...

- If you add one value, you need to add the other value too.
- If you edit one value, you need to edit the other value too.
- If you remove one value, you need to remove the other value too.

User List

The **User List** type supports single-select or multi-select user lists.

Number

The **Number** type supports integer or decimal numeric values. It has the following fields:

- **Minimum value:** represents the smallest value that you can enter into the field.
- **Maximum value:** represents the largest value that you can enter into the field.
- **Decimal places:** used to define how many values to the right of the decimal point are valid. Select 0 for an integer value or 1, 2, 3, or 4 for a decimal value.



Note: Attributes and custom requirement properties display their current value if an update is made to their definition (for example, changes to **Minimum value** or **Maximum value**). To apply the new definition, open and save the asset.

Text

The **Text** type supports any alphanumeric value. It has the following fields:

- **Maximum length:** represents the maximum number of characters that can be entered into the field. Enter 0 or leave this field empty for unrestricted length (for requirement properties only).

Editing Attributes

To edit an attribute:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Attributes** tab to view the list of current attributes.
3. Click the name of the attribute that you want to edit. The **Edit Attribute** dialog box displays.
4. Edit the **Name** of the attribute.

When the attribute is available for use, the name will display in list boxes for filters, execution plans, and tests. Attributes can be used in global filters for filtering by attributes and can be applied to tests and execution plans.

5. Edit the **Description** of the attribute.

- See the topic *Types for Attributes and Custom Requirements Properties* for descriptions of each type of attribute. You have different editing options depending on the attribute data type.
- Optionally check the **Required (Tests)** check box. Required attributes must be set when creating a new test. Required attributes that are not yet set display with an orange background and a red border on the **New Test** dialog.

 **Note:** If the parent test has a certain attribute already set, the value is automatically filled in for the child test. However, the **Inherit from parent** check box is always unchecked by default. If the parent test does not require a certain attribute, the **Inherit from parent** check box is disabled.

 **Note:** If an attribute is set to **Inactive**, it is not required when creating a new test.

- Optionally check the **Required (Execution Planning)** check box. Required attributes must be set when creating a new execution plan, configuration suite, or testing cycle. Required attributes that are not yet set display with an orange background and a red border on the creation dialog.

 **Note:** If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.

 **Note:** If an attribute is set to **Inactive**, it is not required when creating a new element.

- Click **OK**. You are returned to the **Attributes** list.

Deleting Attributes

To delete an attribute:

- In the menu, click **Project:<Project Name> > Project Settings** .
- Click the **Attributes** tab to view the list of current attributes.
- Before you can delete an attribute, you must first deactivate it. In the **Status** column, click the **Active** link or icon and then click **Yes** on the confirmation dialog box to deactivate the attribute.
- Once the attribute is inactive, click **Delete Attribute** to remove it. A confirmation dialog box displays, asking you to confirm the deletion.
- Click **Yes** to remove the selected attribute; or click **No** to abort the operation. If you click **Yes** you will be returned to the **Attributes** page, where the removed attribute is no longer displayed.
- If an error displays, ensure that the selected attribute is not applied to any tests or execution plans, or used in any global filters. You can only delete unused attributes.

Attributes Page

Project:<Project Name> > Project Settings > Attributes

The **Attributes** page lists the attributes that have been created for the current project. For each attribute, the **Attributes** page displays the following columns:

| Column | Description |
|---------|---|
| Actions | Available actions that can be performed on the attribute. |
| Name | Name of the attribute. When the attribute is available for use, the name will display in list boxes for filters, execution plans, and tests. Attributes can be used in global filters for filtering by attributes and can be applied to tests and execution plans. Click the Edit icon to edit the name of the attribute. |
| Type | Attribute type. See the topic <i>Types for Attributes and Custom Requirements Properties</i> for descriptions of each type of attribute. |

| Column | Description |
|-------------------------------|--|
| Status | Status of the attribute, active or inactive. |
| Required (Tests) | Shows whether the attribute is required in the Tests unit. Attributes required for Tests must be set when creating a new test. |
| Required (Execution Planning) | Shows whether the attribute is required in the Execution Planning unit. Attributes required for Execution Planning must be set when creating an execution plan, configuration suite, or testing cycle. |
| Reporting Column | The column name of the attribute in the LQM <i>Reporting</i> table. Use this column name to query the selected attribute within the LQM <i>Reporting</i> table. See the database model documentation for detailed information. |
| Created On | When the attribute was created. |
| Created By | User who created the attribute. |
| Changed On | When the attribute was most recently modified. |
| Changed By | User who most recently modified the attribute. |

Data Sources for Data-Driven Tests

Data-driven tests are tests that are derived from values in an existing data source, such as a spreadsheet or a database. The data sources are managed in a project-specific scope.



Tip: To acknowledge changes in your data source to Silk Central, you must synchronize your data source profile with your data source whenever your data source is updated or changed.

Data Source Limits

Data sources have specific limits to prevent Silk Central from getting into an unstable state due to excessive data generation.

Atomic vs. Non-Atomic Tests

For detailed information on the data-driven test types that are available in Silk Central, see [Atomic and Non-Atomic Data-Driven Tests](#).

Limits

Data sources for data-driven tests have the following default limits:

- 100 rows for non-atomic tests
- 1000 rows for atomic tests
- 20000 rows+cells for *XLSX* data sources (20000 is the amount of rows + the amount of cells. For example: If you have 3000 rows with 6 columns, the calculated amount is $3000 + (3000 * 6) = 21000$, so the limit is exceeded.)
- 220 characters for data set names (Excel sheet names/table names)
- 220 characters for column key names
- The concatenated length of all primary keys may not exceed 255 characters in length for JDBC, Excel, and CSV data sources
- 2000 characters per cell (longer strings will be truncated). Note that data-driven Silk Test Classic tests only accept data source strings in quotation marks. If the string length in a cell exceeds 2000 characters, the string is truncated and the trailing quotation mark is missing, which leads to an error.

Modifying the Default Limits

The default limits can be modified, if required. Modify these values with care though to avoid excessive data generation. Before attempting to modify these values, stop the application server and restart it again after you have saved your changes.

The row limits for atomic and non-atomic tests can be configured in the `TmAppServerHomeConf.xml` file. The default path for this file is `C:\Program Files (x86)\Silk\Silk Central 20.6\instance_<instance number>_<instance name>\Conf\AppServer` on the application server. Modify the values of the `Config/DataDriven/MaxRowsAtomic` and `Config/DataDriven/MaxRows` XML tags, respectively.

The XLSX cell limit can be configured in the `configuration.properties` file in the Excel data source plugin (`DS-Excel.zip`). The default path for this file is `C:\Program Files (x86)\Silk\Silk Central 20.6\instance_<instance number>_<instance name>\Plugins` on the application server. Modify the value of the `excel.importcelllimit` property accordingly.

Configuring JDBC Data Sources

 **Caution:** The concatenated length of all primary keys may not exceed 255 characters in length for JDBC data sources.

To configure a JDBC data source:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click **New Data Source** to open the **New Data Source** dialog box.
4. Type a **Name** for the data source.
5. From the **Data source type** list box, select **JDBC**.



Note: If you are setting up an ODBC data source, you need to manually insert your ODBC **Driver class** and **URL**. For example, **Driver class:** `sun.jdbc.odbc.JdbcOdbcDriver`, **URL:** `jdbc:odbc:MyDatabaseName`. You must also set up an ODBC data source in MS Windows in the **Administrative Tools**. For more information, refer to Microsoft Windows Help. If you have your front-end server and your application server on different machines, make sure that the name of your system data source set-up in Microsoft Windows is the same as the ODBC data source. These names are case-sensitive.

6. If you select **JDBC** as the **Data source type**, the **Driver class** field is populated automatically. In the **URL** field, replace the host name value, `<hostname>`, with the name of the computer that is hosting the data source and replace the database name value, `<databasename>`, with the name of the target database.
7. In the **Username** and **Password** fields, enter valid database credentials.
8. *Optional:* If you are working with a database that includes multiple tables, and you want to narrow down the data source to specific tables, you can browse to and select specific tables for inclusion:
 1. Click [...] next to the **Table filter** field.
 2. The **Select Table Filter** dialog box displays. Select the tables that you want included as your data source.
 3. Click **OK**.
9. *Optional:* **Key column** selection is used by tests to define which worksheet columns within a data source are used as primary key. This is helpful if your data source will undergo edits, for example when you add or remove rows within a worksheet. Even if your data source is edited, tests will still be able to identify which columns or rows should be used. Tests created from data-driven data sources use key column values in their names, rather than column numbers.

To configure a key column:

1. Click [...] next to the **Key column** field.
 2. The **Select Key Column** dialog box displays. Select a column from the column list that is to act as a key column.
 3. Click **OK**.
10. Click **OK** on the **New Data Source** dialog box.

Configuring Microsoft Excel or CSV Data Sources

 **Caution:** Excel worksheets using password protection can not be configured as data sources for Silk Central. Turn off a worksheet's password protection to use it as data source for data-driven testing. The concatenated length of all primary keys may not exceed 255 characters in length for Excel and CSV.

To configure a Microsoft Excel or CSV data source:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click **New Data Source** to open the **New Data Source** dialog box.
4. Type a **Name** for the data source.
5. From the **Data source type** list box, select **MS Excel** to configure a Microsoft Excel data source, or select **CSV** to configure a CSV data source.
6. From the **Source control profile** list box, select the pre-configured source control profile that hosts your data file. For detailed information regarding the configuration of source control profiles, see *Source Control Profiles*.
7. Click **Browse** to open the **Select Source Control Path** dialog box. Browse to and select a data source file of the selected type in your source control path.
8. *Optional:* MS Excel only. If you are working with an Excel spreadsheet that includes multiple worksheets, and you want to narrow down the data source to specific worksheets, you can browse to and select specific worksheets for inclusion. To do this:
 1. Click [...] next to the **Worksheet filter** field.
 2. The **Select Worksheet Filter** dialog box displays. Select the worksheets that you want included as your data source.
 3. Click **OK**.
9. *Optional:* **Key column** selection is used by tests to define which worksheet columns within a data source are used as primary key. This is helpful if your data source will undergo edits, for example when you add or remove rows within a worksheet. Even if your data source is edited, tests will still be able to identify which columns or rows should be used. Tests created from data-driven data sources use key column values in their names, rather than column numbers.

 **Note:** MS Excel only: If the data source includes multiple worksheets, only columns with identical names are available to be defined as key columns.

To configure a key column:

1. Click [...] next to the **Key column** field.
 2. The **Select Key Column** dialog box displays. Select a column from the column list that is to act as a key column.
 3. Click **OK**.
10. Click **OK** on the **New Data Source** dialog box.

Downloading Excel Files from a Data Source

 **Note:** Files can not be downloaded from JDBC and ODBC data sources.

To download an Excel file from a data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Download** icon in the **Actions** column of the data source you want to download from. The **File Download** dialog box displays.
4. Click **Open** to open the file immediately, or click **Save** to specify where on your local system you want to save the file to.

Synchronizing Data Sources

You must synchronize a data source each time it is changed or updated, if you want to make Silk Central aware of the changes. Synchronizing a data source propagates recent changes to associated tests.

To synchronize an updated data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Synchronize** icon in the **Actions** column that corresponds to your data source to propagate the updated file to the associated tests.
4. A confirmation dialog box displays, asking you to confirm the synchronization. Click **Yes** to synchronize all tests with the updated data source, or click **No** to abort the synchronization.



Caution: When you synchronize a data source, all running executions depending on this data source are aborted. Results of incomplete tests within these executions are lost.

5. Click **OK** on the success message dialog box.

Uploading Updated Excel Files to a Data Source



Note: Files can not be uploaded to JDBC and ODBC data sources.

To upload an updated Excel file to a data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Upload** icon in the **Actions** column of the data source you want to download from.
4. Click **Browse...** on the **Upload File** dialog box.
5. Select the updated Excel file that you want to replace the currently uploaded Excel file with. Click **Open**.
6. Click **OK** on the **Upload File** dialog box.
7. A confirmation dialog box displays, asking you to confirm the overwriting of the existing file. Click **Yes** to continue.
8. After uploading the updated data source file, another dialog box displays, asking you if you want to synchronize the tests with the updated data source. Click **Yes** to synchronize immediately, or click **No** if you want to synchronize later.



Note: After uploading an updated data source file, you must synchronize the data source so that associated tests are updated.

Deleting Data Sources



Note: Data sources that are being used by tests can not be deleted.

To delete a data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Delete** icon in the **Actions** column of the data source that you want to delete.
4. A confirmation dialog box displays. Click **Yes** to remove the data source, or click **No** to abort the deletion.

Data Sources Configuration Page

Project:<Project Name> > Project Settings > Data Sources

The **Data Sources** page lists all data sources that are configured for the integration of data-driven tests into Silk Central. Click **New Data Source** to create a new data source. For each data source, the page displays the following columns:

| Column | Description |
|------------------------|---|
| Name | The name of the data source as it displays in the Silk Central GUI and in reports. Click the name to modify the data source settings. |
| Type | Data source type: <ul style="list-style-type: none"> • CSV • JDBC • MS Excel |
| File Name | The name of the file that is used as data source, for example <code>datasource.xlsx</code> . |
| Source Control Profile | The name of the source control profile that hosts the data file. |
| Created On | Date when the data source was created. |
| Created By | User who created the data source. |
| Changed On | Date when the data source was last modified. |
| Changed By | User who last modified the data source. |
| Actions | You can perform the following actions on a data source: <ul style="list-style-type: none"> • Delete • Download • Upload • Synchronize |

Filters

Filters provide an efficient means of finding exactly the information you need, while excluding extraneous detail. Filters highlight only those elements that are relevant to your needs, and enable you to quickly sort through requirements, test elements, and execution plans. By defining global filters, you can create complex filter criteria that are available throughout Silk Central without defining filter criteria each time you need to filter a list.

Based on your needs, you can create new filters, edit existing filters, select filters, delete filters, or turn filtering off at the project level. Projects do not contain default filters. You can access and edit filters from the toolbars in the Silk Central units and from the **Project Settings** unit.

 **Note:** Filters are not applied to reports. The **Recent Changes** filter enables you to view project-wide changes and additions that other users have made to tests since your last change acknowledgement. The **Show Changes/Show All** toggle button and the **Acknowledge** button in the **Tests** area help you to find out what changes other users have made. Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Creating Filters

To create a filter:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Filters** tab to view the list of available filters.
3. Click **New Filter**. The **New Filter** dialog box appears.
4. Type a **Name** for the new filter.
This name will be displayed in list boxes when the filter becomes available.
5. Select a **Category** from the list to make the filter available in the **Requirements, Tests, or Execution Planning** area of Silk Central.
6. *Optional:* Type a **Description** for the new filter.
7. *Optional:* Check the **Visible to other users** check box to allow other users to see the filter.
8. *Optional:* Check the **Editable by other users** check box to allow other users to edit the filter.

 **Note:** If **Visible to other users** and **Editable by other users** are checked, the filter is public. To delete non-public (private) filters, you have to be the owner of the filter or you need the **Delete private filters of other users** permission.

9. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.

 **Note:** You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

10. Select a **Property, Operator, and Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|--------------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

11. *Optional:* Click **More** if you want to add more than one filter category to the new filter. Repeat this procedure to define new categories.



Note: If you define more than one filter category, you must define whether the categories need to be fulfilled in addition to the existing categories (AND relationship), or if the filter returns true when one of the filter categories is fulfilled (OR relationship). Select either **AND** or **OR** to define the filter category relationship. You cannot define nested AND, OR relationships.

12. *Optional:* To remove filter categories, click **Fewer**. This removes the last filter category.

13. Click **OK** to save the new filter, or click **Cancel** to abort the operation.

Creating Advanced Filters

Advanced custom filters enable you to combine simple filters to create complex filters that apply multiple filter criteria simultaneously.

To create an advanced custom filter:

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. Click  (**New Filter**) in the toolbar. The **New Filter** dialog box appears.
3. If necessary, click **Advanced** to show the whole dialog box.
4. Click **More** to display a second set of filter-parameter fields with which you can define a second set of filter parameters.
5. Select a logical operator for the application of the filtering queries. For example, with the operator `and` filtered elements must meet both sets of criteria and with the operator `or` filtered elements must meet one, but not both, of the criteria sets.
6. To delete a filter-parameter string, click .
7. To display additional filter-parameter fields and create additional filter queries, click **More**. To remove excess filter-parameter sets, click **Fewer**.

Editing Filters

To edit a filter:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Filters** tab to view the list of available filters.
3. Click the name of the filter you want to edit. The **Edit Filter** dialog box displays.
4. Edit the **Name** and **Description** of the filter.
5. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.



Note: You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

6. Select a **Property**, **Operator**, and **Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|----------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |

| Operator | Description |
|--------------|--|
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

7. Click **OK** to save the edited filter definition.

Copying Filters

Create a new filter by copying an existing one and changing the settings.

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. In the toolbar of the selected area, select a filter from the list box.
3. Click . The **Edit Filter** dialog box opens with the settings of the original filter.
4. Edit the properties of the filter and click **OK**.

Deleting Filters

To delete a filter:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Filters** tab to view the list of available filters.
3. Click **Delete Filter** for the filter that you want to delete. A confirmation dialog box displays, asking you to confirm the deletion.
4. Click **Yes** to delete the selected filter or **No** to abort the operation. If you click **Yes**, you will be returned to the filters list and the deleted filter is no longer displayed.

Global Filter Details

Global filters allow you to quickly sort through elements in a Silk Central area, highlighting only those that are relevant to your needs.

To view the details of a global filter, click **Project:<Project Name> > Project Settings > Filters** in the menu, and click on the name of the filter.

The specific Silk Central area in which a new filter is available is defined by selecting the general filter category.

Each global filter has to have a property, an operator, and a value from the respective list boxes:

- Property** Available properties depend on the filter category. It defines the property for which you are defining a filter setting. If you selected an attribute category, the property list includes custom attributes to query against.
- Operator** Specifies the filter operator. The operator depends on the property type. For example, if you select a property that is based on a string field type, the available operators are = (equals defined value), *not* (differs from the defined value), *contains* (contains the defined value somewhere in the string), and *does not contain* (does not contain the defined value in the string).
- Value** The value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box.

Recent Changes

The *Recent Changes* filter enables you to efficiently view and acknowledge changes and additions that other users have made to requirements, tests, or execution plans project-wide since your last change acknowledgement.

The two buttons at the far-right of the toolbar, the **Show Changes/Show All** toggle button and the **Acknowledge** button, help you to find out what changes other users have made.

 **Note:** Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Applying Filters

After you have created and stored a custom filter, you can apply that filter to the selected tree. Custom filters can be applied for requirements, tests and execution plans. Only elements that meet applied filter criteria are displayed in the tree.

 **Note:** Filtered requirements are returned in read-only form and can not be edited. The **Edit Properties** button is disabled for filtered requirements.

To apply a stored filter:

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. Select the desired filter from the **Filter** list box on the toolbar.
3. All elements that meet the filter's criteria are displayed.

 **Note:** To remove filtering and display all elements, select **<No Filter>** from the **Filter** list box on the toolbar.

Change Notification

Silk Central can notify you by email when requirements or tests are changed by other users. Each user has the option of activating change-notification. Once notification has been enabled, perform an initial acknowledgment. From then on, you can view and acknowledge changes that have occurred since your last acknowledgment. To avoid numerous notifications, only a single email alert is sent to you when a change is made, regardless of how many changes other users may have made since your last acknowledgment. Email alerts include links that take you directly to a view of recent changes.

Before you can activate change notification for requirements or tests, you must configure your email address in the user settings of Silk Central.

 **Note:** Change notification only works if an email server has been configured by your administrator. If change notification has not been enabled, please contact your Silk Central administrator.

Enabling Change Notification

To enable change notification:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Notifications** tab.
3. Click **Configure Email Notification** to open the **Configure Email Notification** dialog box.
4. If you want to be notified by email when changes are made to requirements in the currently selected project, check the **Changes on Requirements** check box.
5. If you want to be notified by email when changes are made to tests within the currently selected project, check the **Changes on Tests** check box.
6. Click **OK** to save the notification settings, or click **Cancel** to abort the operation without saving changes.

You will be notified by email about changes for which you have activated notification.

Disabling Change Notification

To disable change notification:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Notifications** tab.
3. Click **Configure Email Notification** to open the **Configure Email Notification** dialog box.
4. If you do not want to be notified by email when changes are made to requirements in the currently selected project, uncheck the **Changes on Requirements** check box.
5. If you do not want to be notified by email when changes are made to tests in the currently selected project, uncheck the **Changes on Tests** check box.
6. Click **OK** to save the notification settings or click **Cancel** to abort the operation without saving changes.

Changes Triggering Change Notification

When you have activated change notification to inform you of changes that are made to requirements or tests an email alert is sent to you, following your logout. The email alert is sent to you if one or more of the following settings are changed:

| Area | Changes |
|---------------------|--|
| Requirements | <ul style="list-style-type: none">• A requirement is created or deleted.• The name or description of a requirement is edited.• A system property is edited.• A requirement is set as obsolete.• A requirement is recovered.• A test is assigned to or removed from a requirement.• A custom property of a requirement is created, edited, or deleted. |
| Tests | <ul style="list-style-type: none">• A container is created or edited.• A product is edited.• A source control profile is edited.• "Clear working folder" is edited.• The root node is edited.• The custom data directory is edited.• The include directory is edited.• The hidden test properties are edited.• The Silk Test Classic interface is edited.• A test folder is created, edited, or deleted.• A test is created, edited, or deleted.• The planned time is edited.• A test step is added or edited. |

Notifications Page

Project:<Project Name> > Project Settings > Notifications

The **Notifications** page lists the notification events for the active project. For each notification event, the **Notifications** page displays the following columns:

| Column | Description |
|---------------------|---|
| Notification Events | Name of the notification event that has been set up for the active project. |
| Status | Status of the notification event. Active or inactive. |

Click the **Configure...** button in each section to edit the values.

Email

When an email notification event is activated, a notification email is sent to the user that activated the event, the first time one of the specified settings is changed.

 **Note:** You must have specified an email address to be able to receive email notifications.

Requirement Flagging

You can configure notifications differently for internal and external requirements. All requirement flagging is enabled by default. For more information, see the *Flags* topic.

Structuring Your Test Assets

Silk Central tracks results of test executions in the context of versions and builds of one specific product. Therefore a **Product** needs to be defined before tests can be created.

For automated tests, another requirement for test creation is a **Source Control Profile**. Automated tests in Silk Central actually represent test scripts that are located in source control systems, which are accessed via such source control profiles. Both product and source control profile information are specified on **Test Container** level, which therefore represents the top-hierarchy node of a test assets structure. Within a test container you can then use folders to structure your test assets according to your testing needs and your process.

Remember that although you can have multiple test containers, one execution plan can execute tests of only one test container due to the test result/build/version relation. Therefore, design your test asset structure according to which tests you want to execute together within one execution plan.

Test Containers

Test containers are used to structure your test assets at the top level. Silk Central associates a defined source control profile and a product per container, which impacts what you can do with tests within a specific container.

When you reach the stage of planning your test executions, only the tests of one specific container are available, so you may consider keeping the amount of specified containers as low as possible.

Adding Test Containers

 **Note:** We recommend to use test containers with caution, because they bind the product to the source control profile.

To add a new test container:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the root node to which you want to add the new container as a sub-node.

 **Note:** Containers cannot be nested, but you can add sub-folders to containers.

3. Click  on the toolbar or right-click on the node and choose **New Test Container**.

You can also right-click on the node and choose **New Test Container**.

A new container root node is added to the **Tests** tree and the **New Test Container** dialog box displays.

4. Type a **Name** and **Description** for the container.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. Select any predefined **Product** that is to be associated with this test container from the list box. For details on adding a product profile, see *Adding Products*.
6. Select a predefined source control profile from the **Source Control profile** list box.

If you do not intend to check out any test sources for the test container, for example if you only have manual tests, select ---.



Note: The Silk Central execution servers retrieve the program sources for the tests included in the container from the defined source control profile. Depending on the specified path in your source control profile, you can specify to use a sub-node in the **Root node** field of the test container.

7. *Optional:* Check the **Clear working folder before a test is executed** check box.

If checked, files that have been removed from the source control will also be removed from the working folder. Default is checked.

8. *Optional:* Click **Browse** to select the root node within the selected source control profile.

Test sources are within this root node. The root node is used on the execution servers to retrieve the sources of a test.

9. *Optional:* When Silk Central is integrated with Silk Performer, you can define the **Custom data directory** and **Custom include directory**.

In Silk Performer, the `Include` directory is divided into a `System Include` directory and a `Custom Include` directory. The `Data` directory is divided into a `System Data` directory and a `Custom Data` directory. For additional information, refer to the Silk Performer documentation.

10. The **Hidden Test Properties** portion of the dialog box allows you to specify the test property types that are to be displayed on the test container's **Properties** page, and the **Properties** pages of all test folders within the container.

These settings do not affect the display of individual tests.

11. Click **Edit** to adjust the hidden test property settings. The **Hidden Test Properties** dialog box displays.

12. Uncheck the check boxes of all test types for which you want to have properties displayed.

13. Click **OK** to save your settings and close the **Hidden Test Properties** dialog box.

14. Click **Save** to save your settings.

The new container is added to the **Tests** tree.

Adding Links to Containers

In the **Tests** tree, you can add a link to a test container in the same project. Linked test containers are displayed in read-only mode at the position in the tree where the link is inserted.

To add a link to a test container:

1. In the menu, click **Tests > Details View**.
2. Right-click the node in the **Tests** tree under which you want the linked test container to display.
3. Choose **New Link** if you want to link the test container at the hierarchy level of the selected node, or choose **New Child Link** to link the test container one level below the selected node. The **Select Test Container For Linking** dialog box displays.
4. Select the test container that you want to reference.

 **Note:** If the referenced test container and the container in which you add the link use different source-control profiles, a confirmation dialog box displays, asking you if you really want to create the link. Linking inside a test container to another test container with a different source-control profile can lead to problems when you download or execute a test within the linked container. Click **No** if you want to change the custom include directory of one of the containers first, or click **Yes** to create the link anyway.

5. Click **OK** to confirm your selection.

The linked container is placed within the selected container as a read-only entity. Any changes to the original test container will be reflected in the linked container.

Editing Test Containers

To edit the properties of a test container:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the test container that you want to edit.
3. Click  in the toolbar.
You can also right-click on the test container and select **Edit**.
The **Edit Test Container** dialog box displays.
4. Edit the properties of the test container.
5. Click **OK** to apply your changes.

Test Folders

Use folders to structure your tests within a container.

Adding Test Folders

To add a new test folder:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the test element after which you want to add the new folder.
3. Click  on the toolbar.
You can also right-click on the element and choose **New Test Folder** to add the new folder after the selected element or choose **New Child Test Folder** as a sub-node to the selected element.

 **Note:** For test containers you can add test folders only as sub-nodes.

A new folder node is appended to the **Tests** tree and the **New Folder** dialog box displays.

4. Provide a name and description for the folder.

 **Note:** Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. Click **OK** to save your settings.

The new folder is added to the **Tests** tree.

Editing Test Folders

To modify test folder properties:

1. In the menu, click **Tests > Details View** .
2. Select the folder in the tree.

3. Click  in the toolbar.
You can also right-click on the test folder and select **Edit**.
The **Edit Test Folder** dialog box displays.
4. Edit the name and description of the folder as required.
5. Click **OK** to accept your changes.

Sorting Test Folders

To move a folder up or down within the **Tests** tree:

1. In the menu, click **Tests > Details View**.
2. Select the folder in the tree.
3. Click  on the toolbar to move the folder one step up in the tree or  to move the folder one step down.

Test Packages

Test packages provide support for the structure of third-party test types in Silk Central, and consist of a package root as well as an arbitrary hierarchy of suite nodes and test nodes.

Test packages also provide users with detailed information about a test execution run. Test packages, suite nodes, and test nodes can be individually assigned, along with their issues and attachments, to requirements. This functionality is similar to the functionality of every other test. After a third-party test is converted into a test package, all tests contained in the package can be run individually. Test nodes and suite nodes contained in a test package are provided with an additional property, the **External ID**.

An advantage of test packages is that the structure can be maintained automatically with every test execution. The structure of a test package can be updated according to the results of its runs. The file `<Silk Central installation folder>\wwwroot\silkroot\xsl\output.xsd` contains an XML schema for the structure of the output XML files of test packages.

Test packages enable the entire functionality of the individual tests, with the following exceptions:

- Test containers that contain test packages cannot be linked.
- Test packages cannot be data-driven because they do not possess data-driven properties.
- All success conditions except the execution time-out are disabled and hidden for test package nodes.



Note: Silk Performer tests, Silk Test Classic tests, and manual tests cannot be converted to test packages, as the structure of these tests is supported in Silk Central by default.

The information that provides the internal structure of a test package is specific, and is no longer consistent when you copy the test package. To reuse a test package, you have to copy the test package, revert the copied package into the parent test, apply the test to your needs, and then convert the changed test to a test package.

Creating a Test Package

To create a new test package out of a third-party test:

1. Run the test once to create the `output.xml` file, which contains the structure of the test package.
For additional information on executing a test, see *Executing a Trial Run of a Test*.
2. In the menu, click **Tests > Details View**.
3. In the **Tests** tree, right-click the test and choose **Convert to Test Package**. The selected test is converted to a hierarchy representing the structure of the last execution result.

Using External IDs

You can use **External IDs** to uniquely identify test nodes and suite nodes in test packages. An **External ID** is provided as a property for each test node and each suite node. The automatically generated **External ID**

identifies a unique test method by the fully qualified name of the class and the method with an "~" prepended.

For JUnit tests, the following schema is used for the automatically generated **External ID**: ~<package name>.<class name>#<method name>.

When refactoring JUnit test classes, the automatic generation of the **External ID** is not applicable, because the result information of tests previous to the refactoring will be lost when creating a new test. In this case the **External ID** for the test must be manually defined. The refactored method is re-identifiable, because the **External ID** remains unchanged while moving a JUnit test or changing its name. The **External ID** can be manually set in the source code as an annotation.

The following code example shows such an annotation for JUnit tests:

```
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;

@Retention(RetentionPolicy.RUNTIME)
public @interface ExternalId {
    String externalId();
}
```

The annotation can be used in a JUnit test to annotate classes and test methods as shown:

```
import static org.junit.Assert.*;
import org.junit.Test;

import com.borland.runner.ExternalId;

@ExternalId(externalId="JUnit4test")
public class JUnit4test {

    @Test
    @ExternalId(externalId="MyExtId1")
    public void test1() {
        ...
    }

    @Test
    @ExternalId(externalId="MyExtId2")
    public void test2() {
        ...
    }
}
```

Be aware that using **External IDs** with JUnit runner 'org.junit.runners.Parameterized' is not supported for test methods, because the **External ID** is not unique for repeated runs of a method with different parameters. As a work around an **External ID** could be specified on class level, but must be omitted on method level. An example follows:

```
@RunWith(Parameterized.class)
@ExternalId(externalId="parameterizedWithExtId")
public class TestCaseParameterizedWithExternalId {

    @Parameters
    public static Collection<Object[]> parameterFeeder() {
        return Arrays.asList(new Object[][] {
            { "param_name1", "param_value1" }, // set of parameters per run, type
            matching constructor must exist!
            { "param_name3", "param_value3" },
            { "param_name2", "param_value2" },
        });
    }
}
```

```

private String paramName;
private String paramValue;

public TestCaseParameterizedWithExternalId(String paramName, String
paramValue) {
    this.paramName = paramName;
    this.paramValue = paramValue;
}

@Test
public void testWithParams() {
    System.out.println(String.format("run with parameter: name='%s',
value='%s'", paramName, paramValue));
}
}

```



Note: The setting of the **External ID** for a JUnit test is only possible for tests using JUnit 4.4 or higher.

Reusing a Test Package

The information that provides the internal structure of a test package is specific, and is no longer consistent when you copy the test package. To reuse a test package, you have to copy the test package, revert the copied package into the parent test, apply the test to your needs, and then convert the changed test to a test package.

To reuse the information in a test package:

1. In the menu, click **Tests > Details View**.
2. Right-click the test package that you want to reuse in the **Tests** tree and select **Copy**.
3. Select the node in the **Tests** tree to which you want to add the copied test package.
4. Right-click the selected node and select **Paste**, to add the copied package to the same level in the tree, or select **Paste as Child** to add the copied test package as a sub-node to the selected node.



Caution: The information now contained in the copied test package is not consistent.

5. Right-click the copied package and select **Revert Package to Test**. The package information is removed from the parent test.
6. Edit the test to apply it to your needs.
For additional information, see *Editing Tests*.
7. Right-click on the edited test and select **Convert to Test Package**.

You now have a copy of the original test package with different properties.

Reverting a Test Package to the Test

Revert a test package to the original test to create a different test package out of the parent node.



Note: Reverting a test package to a test hides run result information at the test level. However, you can still see this information in the timeline view or a result file.

To revert a test package to the original test:

1. In the menu, click **Tests > Details View**.
2. Right-click the test package in the **Tests** tree.
3. Click **Revert Package to Test**.

The test package is reverted to the original test.

Cleaning Up a Test Package

To clean up a test package:

1. In the menu, click **Tests > Details View** .
2. Right-click the test package in the **Tests** tree.
3. Click **Clean Up Test Package**.

All tests that were not executed during the last execution of the test package are removed from the test package.

Managing Tests

Add a new test to your testing framework or configure existing tests.

Test Types

Silk Central supports the following test types out-of-the-box:

Manual Tests

Manual tests are step-by-step instructions for testers. Silk Central provides a UI to efficiently execute manual tests, including the option to record videos and capture screen images during test execution. If desired, manual tests can later be converted to automated tests.

Creating or Editing Manual Tests

To create or edit a manual test:

1. In the menu, click **Tests > Details View** .
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Manual Test** from the **Type** list box.
6. In the **Planned time** field, type the expected amount of time for this manual test to execute.
7. Click **Finish**.

Adding or Editing Manual Test Steps

Add or edit manual test steps of a test or a shared steps object in the **Steps** page.

To add or edit a manual test step:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the node for which you want to add or edit steps in the **Tests** or **Libraries** tree.
3. Click the **Steps** tab.
4. Click on the test step that you want to edit, or use the icons on the **Steps** toolbar to insert a new step. The details of the test step are shown in the **Edit Step** view.

5. Edit the name, action description, expected results, and step properties of the test step in the corresponding fields.

You can insert values from data sources into manual test steps in the form of parameters (see *Adding a Data Source Value to a Manual Test Step*). Parameters use the syntax `${<name>}`, for example `${username}`. When converting a manual test to a keyword-driven test, these parameters are automatically added to the generated keyword.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

6. Perform one of the following actions to finish editing the test step:
 - Click **Save and New Step** to save your changes and add a new step after the current one.
 - Click **Save** to save your changes.
 - Click **Discard** to discard your changes.

Converting Manual Tests to Automated Tests

You can convert a manual test to one of the supported automated test types, for example keyword-driven, Silk Performer, Silk Test Classic, NUnit, JUnit, and all installed plug-ins. The process carries manual test parameters over to the automated test, and adds automated parameters to the new automated test.

The manual test parameters that are carried over to automated tests are:

- Name
- Description
- Assigned requirements
- Assigned execution plans
- Assigned issues
- Attachments
- Test steps

To convert a manual test to an automated test:

1. In the menu, click **Tests > Details View** .
2. Right-click the manual test in the **Tests** tree and select **Automate with....**
3. Select the automated test type from the context menu.
4. For information on filling out the dialog, see the appropriate topic in *Creating Tests*.

Converting Automated Tests to Manual Tests

You can convert an automated test to a manual test. This is useful for example if you have an automated test that you want to convert to another test type, for instance from Silk Test Classic to Silk4J. In this case you would convert the Silk Test Classic test to a manual test, and then automate it with Silk4J.



Note: Limitations: Automated to manual test conversion only works for standard automated tests, which excludes package nodes, data-driven nodes, and linked nodes.

1. In the menu, click **Tests > Details View** .
2. Right-click the test in the **Tests** tree and select **Convert to Manual Test**.
3. Confirm the message.



Caution: Test type-specific properties are discarded. If an automated test is currently running, test results are discarded after the test run completes.

Generating Business Components from Manual Tests

Generate new business components out of manual tests to reuse the steps in the manual tests for business process testing.

1. In the menu, click **Tests > Details View** .
2. Select one or more manual tests in the **Tests** tree.
3. Right-click your selection and select **Generate Business Component**.
4. Specify how you want to handle the steps in the new business components.
 - Click **Copy the steps from the manual tests** to copy the steps of each selected manual test into a corresponding new business component. Future changes to the steps of the manual test will not affect the steps of the corresponding business component and the other way round.
 - Click **Share and reuse the steps of the manual tests** to copy the steps of each selected manual test into a corresponding new shared steps asset. Each shared steps asset is then used by both the manual test and the new business component. Future changes to a shared steps asset will affect both the manual test and the business component.
5. Choose the location where you want to place the new business components in the **Libraries** tree.
6. Click **OK**. Silk Central creates a new business component for each selected manual test.

Example

For example, suppose you have selected the manual tests *Test 1* and *Test 2*, and you want to create two new business components out of these manual tests.

- If you choose to generate business components out of these tests by copying the steps from the manual tests, Silk Central generates two new components with the names *Test 1* and *Test 2*. The new component *Test 1* has the same steps as the manual test *Test 1*, and the new component *Test 2* has the same steps as the manual test *Test 2*.
- If you choose to generate business components and to reuse the steps of the manual tests, Silk Central performs the following actions:
 - It generates two new shared steps with the names *Test 1* and *Test 2*.
 - It replaces the steps in the manual test *Test 1* with the new shared step *Test 1*.
 - It replaces the steps in the manual test *Test 2* with the new shared step *Test 2*.
 - It generates two new business components with the names *Test 1* and *Test 2*. The new business component *Test 1* has the new shared step *Test 1* and the new business component *Test 2* has the new shared step *Test 2*.

Importing Manual Tests from an Excel File

Import manual tests from a Microsoft Excel file to add them to the **Tests** area.

1. In the menu, click **Tests > Details View** .
 2. In the **Tests** tree, right-click the folder or container that you want to use as the parent node of the imported test and choose **Import from Excel File**. The **Import Tests from Excel File** dialog box opens.
 3. Click **Manual Tests**.
 4. Select the mapping file from the **Choose Mapping File** list box.
The default mapping file is `TestPropertiesDefaultMapping.xml`. For information about creating customized mapping files, see *Editing the Test Mapping File*.
 5. *Optional:* Click  to edit the mapping file. Click  to delete the mapping file.
-  **Note:** These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.
6. Check the **One-Time Import** check box to import the tests directly under the selected node without creating a common root folder.
 7. Either type the fully qualified path of the test file into the **Test File** field or click **Choose File** to search for the test file.

8. Click **OK**.

Steps Page

Tests > Details View > <Manual Test> > Steps

Tests > Libraries > <Shared Steps Object> > Steps

Tests > Libraries > <Business Component> > Steps

The **Steps** page lists all manual test steps that are included in the selected test, shared steps object, or business component. The page supports standard Windows Explorer style multi-select functionality.

To create a new shared steps object, select the steps you want to share and click .

The page includes the following toolbar items:

| Toolbar Item | Description | | | | | | | | |
|---|--|--------------------|---|--------------------|--|---------------------------|--|---------------|--------------------------|
|  | Add a new test step to the end of the list. | | | | | | | | |
|  | Insert a new test step before the selected step. | | | | | | | | |
|  | Add a call to a shared steps object from a library to the end of the list. | | | | | | | | |
|  | Add a call to a shared steps object from a library before the selected step. | | | | | | | | |
|  | Delete the selected test steps from the list. | | | | | | | | |
|  | Cut the selected test step from the list and move it to the clipboard. | | | | | | | | |
|  | Copy the selected test steps to the clipboard. | | | | | | | | |
|  | Paste a copy of the test steps held on the clipboard to the row above the selected in the list. | | | | | | | | |
|  | Move the selected test step one position up in the list. | | | | | | | | |
|  | Moves the selected test step one position down in the list. | | | | | | | | |
|  | Opens the Attachments dialog box, where you can perform the following actions: <table border="0"><tr><td>Upload File</td><td>Upload and attach a file to the selected test step.</td></tr><tr><td>Attach Link</td><td>Attach a link to the selected test step.</td></tr><tr><td>Download All Files</td><td>Download all attached files from the selected test step.</td></tr><tr><td>Delete</td><td>Delete the file or link.</td></tr></table> | Upload File | Upload and attach a file to the selected test step. | Attach Link | Attach a link to the selected test step. | Download All Files | Download all attached files from the selected test step. | Delete | Delete the file or link. |
| Upload File | Upload and attach a file to the selected test step. | | | | | | | | |
| Attach Link | Attach a link to the selected test step. | | | | | | | | |
| Download All Files | Download all attached files from the selected test step. | | | | | | | | |
| Delete | Delete the file or link. | | | | | | | | |
|  | When viewing a called shared steps object, click here to view the original steps. The resolved steps are marked with an arrow, and you can access them by clicking on their name. You cannot edit steps while this view is enabled. | | | | | | | | |
|  | View steps called from a shared steps object in a library as a single step. Enable this view to edit test steps. | | | | | | | | |
|  | Create a shared steps object out of the selected steps. | | | | | | | | |
|  | Extract calls from other shared steps objects. Copies all steps in the calls you have selected in the grid into the manual test, shared steps object, or business component. | | | | | | | | |

The **Steps** page supports the following keyboard functions (shortcuts) for test elements:

| Keys | Action |
|---------------|------------|
| - | Move up |
| + | Move down |
| Ctrl+A | Select all |
| Ctrl+C | Copy |
| Ctrl+X | Cut |
| Ctrl+V | Paste |
| Ins | Insert |
| Del | Delete |

The steps are displayed in a grid with the following columns:

| Column | Description |
|-----------------------|--|
| Order | Number of the step in the execution sequence. |
| Name | Name of the test step. |
| Action Description | Action you must perform to execute the test step. |
| Expected Results | Expected result of the test step. |
| Attachments | Amount of links or files that are attached to the test step. |

The **Edit Step** view shows the details of the selected test step and allows you to edit the name, action description, expected results, and step properties of the test step.

The following buttons are available in the **Edit Step** view:

| Button | Description |
|--------------------------|---|
| Save and New Step | Click when you are finished with editing the current step and you wish to save your changes and add a new step after the current. |
| Save | Click when you are finished with editing the current step and you wish to save your changes. |
| Discard | Click to discard your changes. |

Test Asset Workflow

Managing and coordinating existing company processes with an external software tool can be challenging. Silk Central's workflow feature lets you process manual tests in a workflow to map your corporate environment.

The workflow feature allows you to set a state for each test asset to define what tasks can be performed on the asset. By using workflow states, you can resolve frequent issues, like tests that are already being executed although they are still in a design phase.

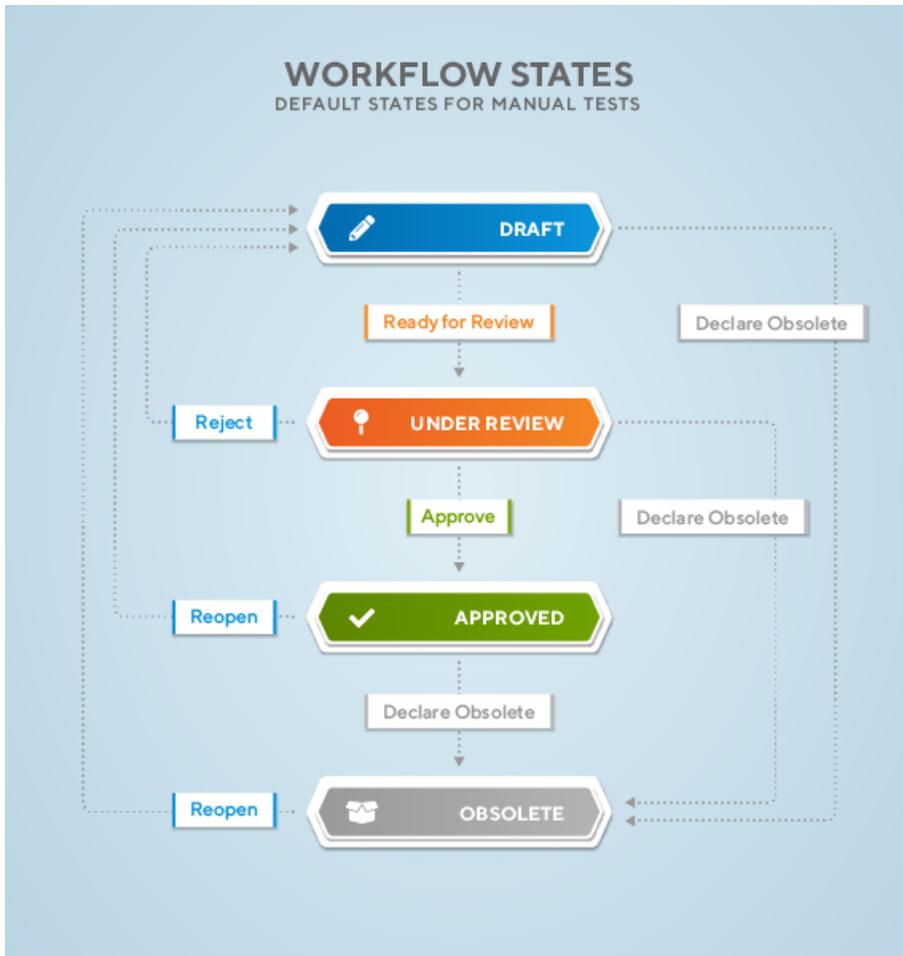
As an extended capability, Silk Central's test asset workflow feature simplifies the communication within big teams and allows them to collaborate efficiently.

You can assign a workflow state to each manual test.

The following default workflow states are available:

| | | | | |
|-----------|--------------|---------------------|-----------------|-----------------|
| to / from | Draft | Under Review | Approved | Obsolete |
|-----------|--------------|---------------------|-----------------|-----------------|

| | | | | |
|---------------------|------------------|------------------|------------------|--------|
| Draft | | Reject | Reopen | Reopen |
| Under Review | Ready for Review | | | |
| Approved | | Approve | | |
| Obsolete | Declare Obsolete | Declare Obsolete | Declare Obsolete | |



When working with workflows, note the following specifics:

- The initial state of a newly created test asset is **Draft**.
- If you click, for example, **Ready for Review**, the state of the test asset changes from **Draft** to **Under Review** and a new version is generated.
- Resetting the state to **Draft** will not create a new version.
- Setting the state again to **Under Review** will create another version.
- If you change the state of a test asset that contains shared steps, the shared steps will be extracted and included in the new version.
- If you revert to such a version, the version will contain the extracted steps and not the Shared Steps.

You can perform the following actions on workflow states:

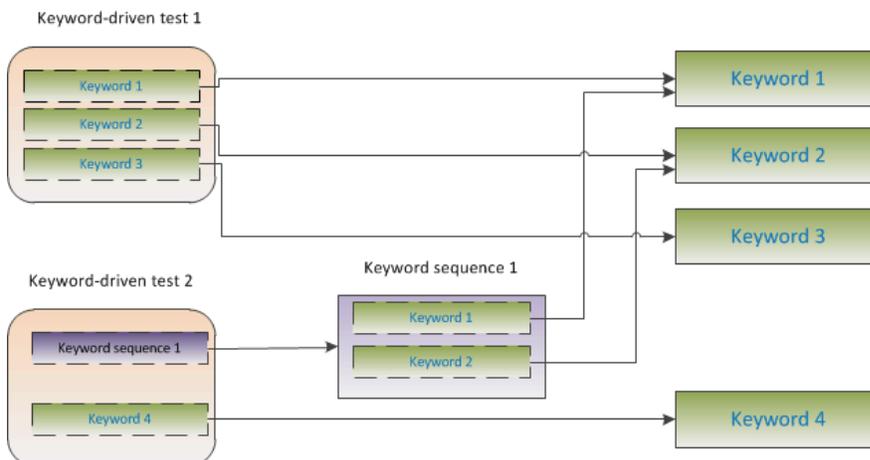
- To change a workflow state, right-click a test asset in the tree of the **Details View** or the **Document View**, go to **Workflow** and click one of the workflow actions (for example **Approve**).
- Approving a test asset requires the **Approve tests** permission. By default, the user role **Test Manager** owns this permission. For more information, refer to [Tests and Executions Permissions](#).
- To show the current workflow state, click a test asset in the **Details View**. The **Workflow State** displays on the **Properties** tab, within the **Details** section.

- An asterisk displaying beside a workflow state (for example `Approved*`) means that changes have been made to the test asset since the current workflow state was set. To view all changes and versions of a test asset, click the **History** tab.
- You can filter for individual workflow states in the **Grid View**.

Keyword-Driven Tests

Keyword-driven testing is a software testing methodology that separates test design from test development and therefore allows the involvement of additional professional groups, for example business analysts, in the test automation process. Silk Central and Silk Test support the keyword-driven testing methodology and allow a very close collaboration between automation engineers and business analysts by having automation engineers develop a maintainable automation framework consisting of shared assets in the form of keywords in Silk Test. These keywords can then be used by business analysts either in Silk Test to create new keyword-driven tests or in Silk Central to convert their existing manual test assets to automated tests or to create new keyword-driven tests.

- A *keyword-driven test* is an executable collection of keywords. A keyword-driven test can be played back just like any other test.
- A *keyword sequence* is a keyword that is a combination of other keywords. Keyword sequences bundle often encountered combinations of keywords into a single keyword, enabling you to reduce maintenance effort and to keep your tests well-arranged.
- A *keyword* is a defined combination of one or more actions on a test object. The implementation of a keyword can be done with various tools and programming languages, for example Java or .NET.



There are two phases required to create keyword-driven tests:

1. Designing the test.
2. Implementing the keywords.

Advantages of Keyword-Driven Testing

The advantages of using the keyword-driven testing methodology are the following:

- Keyword-driven testing separates test automation from test case design, which allows for better division of labor and collaboration between test engineers implementing keywords and subject matter experts designing test cases.
- Tests can be developed early, without requiring access to the application under test, and the keywords can be implemented later.
- Tests can be developed without programming knowledge.
- Keyword-driven tests require less maintenance in the long run. You need to maintain the keywords, and all keyword-driven tests using these keywords are automatically updated.
- Test cases are concise.

- Test cases are easier to read and to understand for a non-technical audience.
- Test cases are easy to modify.
- New test cases can reuse existing keywords, which amongst else makes it easier to achieve a greater test coverage.
- The internal complexity of the keyword implementation is not visible to a user that needs to create or execute a keyword-driven test.

Keywords

A *keyword* is a defined combination of one or more actions on a test object. The implementation of a keyword can be done with various tools and programming languages, for example Java or .NET. In Silk Test, a keyword is an annotated test method (`@Keyword`). Keywords are saved as keyword assets.

Keyword-driven tests and the associated keywords can be designed and managed in Silk Central, while the actual implementation is done in a functional testing tool. Silk Central and Silk Test support the keyword-driven testing methodology and allow a very close collaboration between automation engineers and business analysts by having automation engineers develop a maintainable automation framework consisting of shared assets in the form of keywords in Silk Test. These keywords can then be used by business analysts either in Silk Test to create new keyword-driven tests or in Silk Central to convert their existing manual test assets to automated tests or to create new keyword-driven tests.

A *keyword sequence* is a keyword that is a combination of other keywords. Keyword sequences bundle often encountered combinations of keywords into a single keyword, enabling you to reduce maintenance effort and to keep your tests well-arranged.

Test Automation Overview

Test automation is the process of using software to control the execution of tests by using verifications to compare the *actual* outcomes of the execution with the *expected* outcomes.

A *test automation framework* is an integrated system that sets the rules of automation of a specific product. This system integrates the tests, the test data sources, the object details, and various reusable modules. These components need to be assembled to represent a business process. The framework provides the basis of test automation and simplifies the automation effort.

Creating a Keyword-Driven Test in Silk Central

1. In the menu, click **Tests > Details View** .
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. Select the test type **Keyword-Driven Test** from the **Type** list.
6. Select the **Library** which contains the keywords that you want to use.
The library is required to store the set of keywords that the keyword-driven test uses.
7. Click **Finish**.

Creating a Keyword-Driven Test by Automating a Manual Test

To convert a manual test to a keyword-driven test in Silk Central:

1. In the menu, click **Tests > Details View** .
2. Right-click the manual test in the **Tests** tree and select **Automate with... > Keyword-Driven Test** from the context menu. The **Keyword-Driven Test Properties** dialog box appears.
3. Select the **Library** which contains the keywords that you want to use.
The library is required to store the set of keywords that the keyword-driven test uses.
4. Click **Finish**.

Automating a manual test as keyword-driven test generates the following:

- The manual test is converted to an automated keyword-driven test, containing a draft keyword for each test step. These keywords are added to the library that has been selected for this test.
- If the test contains calls to shared steps that already exist as keyword sequences, these keyword sequences are referenced.
- If the test contains calls to shared steps that don't exist as keyword sequences, these shared steps objects are referenced and enhanced to also be keyword sequences, containing a draft keyword for each test step. This also applies to nested shared steps (shared steps that reference other shared steps). If the hierarchy of nested shared steps is more than 30 levels deep, draft keywords for calls to shared steps below that level are created.
- If the test contains calls to shared steps that are in a different library, draft keywords for these steps are created and referenced.

If your test steps contain parameters that you want to access in the generated keywords, the **Action Description** of a test step must use the correct syntax. Parameters use the syntax `${<name>}`, for example `${username}`. When converting a manual test to a keyword-driven test, these parameters are automatically added to the generated keyword.

Managing Keywords in a Test

Tests > Details View > <Test> > Keywords

The **Keywords** page enables you to manage the keywords of the selected keyword-driven test. The following actions are possible:

| Task | Steps |
|---|---|
| Opening a test or keyword sequence in Silk Test | Click Open with Silk Test to open the selected test or keyword sequence in Silk Test. |
| Adding a keyword | <ol style="list-style-type: none"> 1. Click New Keyword at the bottom of the keywords list, or right-click a keyword and select Insert Keyword Above from the context menu.  Note: You can let Silk Central recommend keywords based on their usage. Toggle the recommendations on or off with Enable Recommendations or Disable Recommendations in the context menu. For additional information, see <i>Which Keywords Does Silk Central Recommend?</i> 2. Select a keyword from the list of available keywords or type a new name to create a new keyword. 3. Click Save. <p>Alternatively, double click an existing keyword in the All Keywords pane on the right or drag and drop it.</p>  Tip: You can select multiple keywords with Ctrl+Click . When dropping them, they will be sorted in the order that you selected them in. |
| Deleting a keyword | Click  in the Actions column of the keyword that you want to delete. Click Save . |

| Task | Steps |
|--|--|
| Changing the order of keywords | Drag and drop a keyword to the desired position. Click Save . |
| Creating a keyword sequence (a keyword consisting of other keywords) | <ol style="list-style-type: none"> 1. Select the keywords that you want to combine in the keywords list. Use Ctrl+Click or Shift+Click on the row number column to select multiple keywords. 2. Right-click your selection and click Combine. 3. Specify a Name and Description for the new keyword sequence. |
| Extracting keywords from a keyword sequence | Right-click a keyword sequence and click Extract keywords . The original keyword sequence is then replaced by the keywords that it contained, but it is not removed from the library. Click Save . |
| Copying and pasting keywords into tests or keyword sequences | <ol style="list-style-type: none"> 1. Select the keywords that you want to copy in the keywords list. Use Ctrl+Click or Shift+Click on the row number column to select multiple keywords. 2. Press Ctrl+C to copy your selection, or Ctrl+X if you want to move the keywords. 3. Open the test or keyword sequence that you want to copy the keywords to and select the row above which the keywords will be inserted. 4. Press Ctrl+V. <p> Tip: You can also paste your selected keywords into Excel, and copy and paste them from there into your tests or keyword sequences.</p> |
| Defining parameters for a keyword sequence | <ol style="list-style-type: none"> 1. Click Parameters above the keywords list. The Parameters dialog box appears. 2. Click Add Parameter. 3. Specify a Name for the new parameter. If the parameter is an outgoing parameter (delivers a value, instead of requiring an input value), check the Output checkbox. 4. Click OK. 5. Click Save. <p> Note: A keyword or a keyword sequence can have a combined total of 20 input and output parameters.</p> |
| Editing a draft keyword | <ol style="list-style-type: none"> 1. Click  in the Actions column of the draft keyword that you want to edit. 2. Select a Group or specify a new group for the keyword. 3. Type a Description for the keyword. This information is valuable for the engineer who will implement the keyword. 4. Click OK. 5. <i>Optional:</i> Click into a parameter field to add parameters for the keyword. If the keyword is implemented with Silk Test, these parameters will appear in the generated code stub. 6. Click Save. |
| Searching for a keyword | <p>Use the search field in the All Keywords pane on the right to find a specific keyword. When you enter alphanumeric characters, the list is dynamically updated with all existing matches. Tips for searching:</p> <ul style="list-style-type: none"> • The search is case-insensitive: doAction will find doaction and DOAction. |

| Task | Steps |
|---------------------------------------|--|
| | <ul style="list-style-type: none"> • Enter only capital letters to perform a so-called <i>Camel/Case</i> search: ECD will find Enter Car Details, Enter Contact Details and EnterContactDetails. • Keyword and group names are considered: test will find all keywords that contain test and all keywords in groups where the group name contains test. • ? replaces 0-1 characters: user?test will find userTest and usersTest. • * replaces 0-n characters: my*keyword will find myKeyword, myNewKeyword and my_other_keyword. • <string> . only searches in group names: group . will find all keywords in groups where the group name contains group. • .<string> only searches in keyword names: .keyword will find all keywords that contain keyword. • <string> .<string> searches for a keyword in a specific group: group . word will find myKeyword in the group myGroup. • Use quotes to search for an exact match: 'Keyword' will find Keyword and MyKeyword, but not keyword. |
| Finding draft keywords in the library | If you want to switch to the library that contains the draft keywords of your selected test or keyword sequence, click Draft keywords in the top right of the editor page. |

Managing Keywords in a Library

The **Keywords** page offers two different views in **Tests > Libraries** , depending on the selected object.

Library

If your selected object is a library, all keywords and keyword sequences within this library are listed.

Libraries that contain uploaded keyword libraries are displayed with the symbol . The following actions are possible:

| Task | Steps |
|---|---|
| Implementing a keyword (adding code to be able to actually run the keyword in a test) | <p>Prerequisites: If your keyword contains parameters, you need to specify a value for each parameter before you implement the keyword.</p> <ol style="list-style-type: none"> 1. Click  (Implement with Silk Test) in the Actions column of the keyword that you want to implement. <ul style="list-style-type: none">  Note: If your browser reports a security warning for opening an external application, confirm the message. 2. The Silk Test client opens. Refer to the Silk Test client documentation for detailed information. 3. After having implemented a keyword, upload it back to Silk Central. |
| Editing an implemented keyword in Silk Test | Click  (Open with Silk Test) to open the keyword in the Silk Test client. Refer to the Silk Test client documentation for detailed information. |

| Task | Steps |
|---|--|
| Converting a draft keyword to a keyword sequence (a keyword consisting of other keywords) | Click  in the Actions column of the keyword that you want to convert to a keyword sequence. The original keyword is then replaced by a keyword sequence with the same name and the keyword editor opens. |
| Converting a keyword sequence to a draft keyword | <p>If you accidentally created a keyword sequence, you may want to convert it back to a draft keyword. Click  (Convert to draft keyword) in the Actions column of the keyword sequence that you want to convert.</p> <p> Note: Converting a keyword sequence to a draft will delete the sequence and all contained keywords. This action cannot be undone.</p> |
| Editing a keyword sequence | Click  to open the keyword sequence in the keyword editor. |
| Grouping keywords | <p>To better structure the keywords in a library, you can group them:</p> <ul style="list-style-type: none"> • Silk Central keyword sequences can be placed in sub-folders. The folder names are reflected as group names and your keywords are grouped accordingly in the list. • Keyword groups of implemented keywords are defined in the implemented code. These group names are used by Silk Central and your keywords are grouped accordingly in the list. |
| Viewing tests and keyword sequences that reference a specific keyword or keyword sequence | To view all tests and keyword sequences that reference a specific keyword or keyword sequence, select the appropriate entry in the keyword list and the Referenced By list will display all referencing keyword-driven tests and keyword sequences. Simply click the name to jump to the respective reference. |
| Replacing keywords in referencing tests and keyword sequences | <p>To replace references to the selected keyword or keyword sequence with another one, select the appropriate entry in the keyword list and use the Referenced By list to select the objects in which you want to replace the reference.</p> <ol style="list-style-type: none"> 1. Select the keyword or keyword sequence that you want to replace. 2. In the Referenced By list, use Ctrl+Click or Shift+Click to select the objects in which you want to replace the reference. 3. Right-click the selection and choose Replace keyword in selected reference(s). The Replace References dialog box appears. 4. Select the keyword that you want to replace the current keyword with. 5. If the selected keyword contains parameters, click Next, otherwise click OK and you are done. 6. The parameters of the selected keyword are listed. For each parameter, select the matching parameter of the previous keyword, or enter a value. 7. Click OK. |

Keyword Sequence

If your selected object is a keyword sequence (a keyword consisting of other keywords), the **Keywords** page displays the keyword editor, which enables you to manage the keywords of the selected keyword sequence. The following actions are possible:

| Task | Steps |
|---|--|
| Opening a test or keyword sequence in Silk Test | Click Open with Silk Test to open the selected test or keyword sequence in Silk Test. |

| Task | Steps |
|--|--|
| Adding a keyword | <ol style="list-style-type: none"> 1. Click New Keyword at the bottom of the keywords list, or right-click a keyword and select Insert Keyword Above from the context menu. <ul style="list-style-type: none">  Note: You can let Silk Central recommend keywords based on their usage. Toggle the recommendations on or off with Enable Recommendations or Disable Recommendations in the context menu. For additional information, see <i>Which Keywords Does Silk Central Recommend?</i> 2. Select a keyword from the list of available keywords or type a new name to create a new keyword. 3. Click Save. <p>Alternatively, double click an existing keyword in the All Keywords pane on the right or drag and drop it.</p> <ul style="list-style-type: none">  Tip: You can select multiple keywords with Ctrl+Click. When dropping them, they will be sorted in the order that you selected them in. |
| Deleting a keyword | Click  in the Actions column of the keyword that you want to delete. Click Save . |
| Changing the order of keywords | Drag and drop a keyword to the desired position. Click Save . |
| Creating a keyword sequence (a keyword consisting of other keywords) | <ol style="list-style-type: none"> 1. Select the keywords that you want to combine in the keywords list. Use Ctrl+Click or Shift+Click on the row number column to select multiple keywords. 2. Right-click your selection and click Combine. 3. Specify a Name and Description for the new keyword sequence. |
| Extracting keywords from a keyword sequence | Right-click a keyword sequence and click Extract keywords . The original keyword sequence is then replaced by the keywords that it contained, but it is not removed from the library. Click Save . |
| Defining parameters for a keyword sequence | <ol style="list-style-type: none"> 1. Click Parameters above the keywords list. The Parameters dialog box appears. 2. Click Add Parameter. 3. Specify a Name for the new parameter. If the parameter is an outgoing parameter (delivers a value, instead of requiring an input value), check the Output checkbox. 4. Click OK. 5. Click Save. |
| Editing a draft keyword | 1. Click  in the Actions column of the draft keyword that you want to edit. |

| Task | Steps |
|---------------------------------------|--|
| | <ol style="list-style-type: none"> 2. Select a Group or specify a new group for the keyword. 3. Type a Description for the keyword. This information is valuable for the engineer who will implement the keyword. 4. Click OK. 5. <i>Optional:</i> Click into a parameter field to add parameters for the keyword. If the keyword is implemented with Silk Test, these parameters will appear in the generated code stub. 6. Click Save. |
| Searching for a keyword | <p>Use the search field in the All Keywords pane on the right to find a specific keyword. When you enter alphanumeric characters, the list is dynamically updated with all existing matches. Tips for searching:</p> <ul style="list-style-type: none"> • The search is case-insensitive: <code>doAction</code> will find <code>doaction</code> and <code>DOAction</code>. • Enter only capital letters to perform a so-called <i>CamelCase</i> search: <code>ECD</code> will find <code>Enter Car Details</code>, <code>Enter Contact Details</code> and <code>EnterContactDetails</code>. • Keyword and group names are considered: <code>test</code> will find all keywords that contain <code>test</code> and all keywords in groups where the group name contains <code>test</code>. • <code>?</code> replaces 0-1 characters: <code>user?test</code> will find <code>userTest</code> and <code>usersTest</code>. • <code>*</code> replaces 0-n characters: <code>my*keyword</code> will find <code>myKeyword</code>, <code>myNewKeyword</code> and <code>my_other_keyword</code>. • <code><string>.</code> only searches in group names: <code>group.</code> will find all keywords in groups where the group name contains <code>group</code>. • <code>.<string></code> only searches in keyword names: <code>.keyword</code> will find all keywords that contain <code>keyword</code>. • <code><string>.<string></code> searches for a keyword in a specific group: <code>group.word</code> will find <code>myKeyword</code> in the group <code>myGroup</code>. • Use quotes to search for an exact match: <code>'Keyword'</code> will find <code>Keyword</code> and <code>MyKeyword</code>, but not <code>keyword</code>. |
| Finding draft keywords in the library | <p>If you want to switch to the library that contains the draft keywords of your selected test or keyword sequence, click Draft keywords in the top right of the editor page.</p> |

Cross-Browser Testing with Keyword-Driven Tests

Keyword-driven tests that are implemented with Silk Test and contain a `BrowserBaseState` can be parameterized in Silk Central so that you can run these tests across multiple browsers. To configure the browsers that you want to test:

1. Create a configuration suite and assign the keyword-driven tests that you want to run across multiple browsers. For detailed information about creating a configuration suite, see [Creating a Configuration Suite](#).
2. On the **Configurations** tab, click **Add/Remove Parameters** and create a new parameter with the name `silktest.configurationName`.
3. Click **New Configuration** to add the configurations that you want to test. For each configuration, specify the browser that you want to test by entering one of the following values for the `silktest.configurationName` parameter:
 - Internet Explorer
 - Firefox
 - Chrome
4. Run the configuration suite.

Building a Keyword-Driven Library Using Java

Build an external keyword-driven library with Java to upload it to Silk Central. This enables you to easily integrate your existing test framework into Silk Central, for example if you already have existing Selenium tests.

1. Select **Help > Tools** and download the **Java Keyword Library Tool**.
2. Add the downloaded `com.borland.silk.keyworddriven.jar` to the classpath of your Java project.
3. Create a Java class that will contain your Keywords.
4. Annotate the Java methods that you want to use as keywords. The following sample outlines a class that contains one keyword:

```
import com.borland.silk.keyworddriven.annotations.Argument;
import com.borland.silk.keyworddriven.annotations.Keyword;

public class Keywords {

    @Keyword("My Keyword")
    public void myKeyword(@Argument("Username") String name) {
        System.out.println(name);
    }
}
```

5. Compile your sources and create a Java archive (jar).
If you are using Java 9 or later, ensure that the `Created-By` property in the `MANIFEST.MF` file contains the Java version that the library was built with.
6. Call the command line tool that is contained in the downloaded jar file with the following arguments:
 - `java`
 - `-jar com.borland.silk.keyworddriven.jar`
 - `-build`
 - Library name to be generated.
 - Library type, only Java is currently supported.
 - Binaries path where your keyword library jar file resides and if applicable, any other required libraries.
 - Name of the output library package (zip archive) to be generated.

The following example outlines the command line to build a keyword library:

```
java -jar com.borland.silk.keyworddriven.jar -build
MyLibrary Java ./output "./output/library.zip"
```

Select **Tests > Libraries > <Library> > Properties** . The generated zip archive can be uploaded to Silk Central with the command line tool or in the **Keyword Library** section.

Uploading a Keyword Library from the Command Line

Upload an external keyword library from a Java-based command line to integrate Silk Central and your keyword-driven tests into your continuous integration build system, for example Jenkins.

To upload your keyword library from a Java-based command line:

1. Select **Help > Tools** and download the **Java Keyword Library Tool**.
2. Call the command line tool that is contained in the downloaded jar file with the following arguments:

- java
- -jar com.borland.silk.keyworddriven.jar
- -upload
- Library name of the library in Silk Central to be updated, or created if it does not yet exist.
- Package name of the library package (zip archive) to be uploaded.
- Hostname:port of the Silk Central front-end server.
- Web-service token of the Silk Central user. Required for authentication. You can generate a web-service token in the **User Settings** page of Silk Central, which you can access by clicking on the user name in the Silk Central menu.



Note: For security reasons, Micro Focus recommends using a web-service token for authentication instead of sending your user name and password over the network.

- Username of the Silk Central user. Not required when using a web-service token for authentication.
- Password of the Silk Central user. Not required when using a web-service token for authentication.
- Update information, describing the changes that were applied to the library, in quotes.
- [-allowUsedKeywordDeletion], an optional flag to allow the deletion of keywords that are used in a test or keyword sequence. By default, an error is raised if used keywords are attempted to be deleted.

The following example outlines the command line to upload a library to Silk Central with Java 9 or later:

```
java --add-modules=java.activation,java.xml.ws -jar
com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```

Examples

The following example outlines the command line to upload a library to Silk Central with Java 8 or prior by using a web-service token for authentication:

```
java -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scToken
"Build xy: Implemented missing keywords"
```

To upload the same library with Java 8 or prior by using user name and password for authentication, use a command like the following:

```
java -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```

The corresponding commands with Java 9 or later are:

```
java --add-modules=java.activation,java.xml.ws -jar
com.borland.silk.keyworddriven.jar -upload
```

```
"My library" ".\output\library.zip" silkcentral:19120 scToken
"Build xy: Implemented missing keywords"

java --add-modules=java.activation,java.xml.ws -jar
com.borland.silk.keyworddriven.jar -upload
"My library" ".\output\library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```



Note: When uploading a keyword-driven library with Java 9 or later, ensure `JAVA_HOME` is defined on the execution servers and points to a JDK with the corresponding Java version.

Executing Silk4J/Silk4NET Keyword-Driven Tests Without Uploading a Library

If you have an existing test automation framework consisting of Silk4J or Silk4NET keyword-driven tests and you want to execute them from Silk Central, you can do this by using Silk Central's JUnit and test package functionality. To do so, follow the steps below:

1. Create a source control profile that points to the artifacts of your Silk Test project (see [Source Control Profiles](#)).
2. Create a test container that uses this source control profile (see [Adding Test Containers](#)).
3. Create a JUnit test (see [Creating Tests](#)) and configure the following:
 - For keyword-driven tests that are implemented in Silk4J:
 - Specify the classpath. If you are using the default Silk Test build structure, specify `.\output`.
 - The `.kdt` files need to be in the folder that you specified as classpath, either packaged into a `.jar` file or within a folder in the classpath.
 - For keyword-driven tests that are implemented in Silk4NET:
 - Specify `./Keyword Driven Tests` as classpath.
 - Create a parameter for the test container of your tests (see [Creating Custom Parameters](#)) called `silktest.keyworddriven.dotnetassembly` with a relative path to the .Net Assembly, for example `\bin\Release\Silk4NETProject.dll`.
 - The Silk Test folder called `Keyword Driven Tests` containing the `.kdt` files has to reside in the root folder of the source control.
4. Assign the test to an execution plan (see [Assigned Tests](#)).
5. *Optional:* Once executed, you can right-click the test and select **Convert to Test Package** to get a package structure (see [Creating a Test Package](#)).

Example: Keywords with Parameters

This topic provides an example of how you can use keywords with parameters. A keyword or a keyword sequence can have a combined total of 20 input and output parameters.

As a first step, create a keyword-driven test which contains the keywords that you want to use. You can do this by recording an entire keyword-driven test, or by creating a new keyword-driven test and by adding the keywords in the keyword-driven test editor.

In this example, the keyword-driven test includes the following keywords:

- | | |
|--------------------------|--|
| Start application | This is the standard keyword that starts the AUT and sets the base state. |
| Login | This keyword logs into the AUT with a specific user, identified by a user name and a password. |
| GetCurrentUser | This keyword returns the name of the user that is currently logged in to the AUT. |
| AssertEquals | This keyword compares two values. |

Logout This keyword logs the user out from the AUT.

The next step is to add the parameters to the keywords. To do this, open the test scripts of the keywords and add the parameters to the methods.

To add the input parameters `UserName` and `Password` to the keyword `Login`, change

```
@Keyword("Login")
public void login() {
    ...
}
```

to

```
@Keyword("Login")
public void login(String UserName, String Password) {
    ...
}
```

To add the output parameter `UserName` to the keyword `GetCurrentUser`, change

```
@Keyword("GetCurrentUser")
public void getCurrentUser() {
    ...
}
```

to

```
@Keyword("GetCurrentUser")
public void getCurrentUser(OutParameter CurrentUser) {
    ...
}
```

The keyword-driven test in the **Keyword-Driven Test Editor** should look similar to the following:

| | | Keyword | Parameters | |
|---|---|-------------------|----------------------|-----------------|
| 1 |   | Start application | | |
| 2 |   | Login | <i>UserName</i> | <i>Password</i> |
| 3 |   | GetCurrentUser | <i>CurrentUser ←</i> | |
| 4 |   | AssertEquals | <i>Expected</i> | <i>Actual</i> |
| 5 |   | Logout | | |

Now you can specify actual values for the input parameters in the **Keyword-Driven Test Editor**. To retrieve the value of the output parameter `UserName` of the keyword `GetCurrentUser`, provide a variable, for example `${current user}`. You can then pass the value that is stored in the variable to subsequent keywords.

| | | Keyword | Parameters | |
|---|---|-------------------|-------------------------------|-------------------------------|
| 1 |   | Start application | | |
| 2 |   | Login | <i>UserName</i> | <i>Password</i> |
| 3 |   | GetCurrentUser | <code>\${current user}</code> | |
| 4 |   | AssertEquals | John Smith | <code>\${current user}</code> |
| 5 |   | Logout | | |

Which Keywords Does Silk Central Recommend?

When you add keywords to a keyword-driven test or a keyword sequence in the **Keyword-Driven Test Editor**, Silk Central recommends existing keywords which you might want to use as the next keyword in your test. The recommended keywords are listed on top of the keywords list, and are indicated by a bar graph, with the filled-out portion of the graph corresponding to how much Silk Central recommends the keyword.

Silk Central recommends the keywords based on the following:

- When you add the first keyword to a keyword-driven test or a keyword sequence, Silk Central searches for similar keywords that are used as the first keyword in other keyword-driven tests or keyword sequences. The keywords that are used most frequently are recommended higher.
- When you add additional keywords to a keyword-driven test or a keyword sequence, which already includes other keywords, Silk Central recommends keywords as follows:
 - If there are keywords before the position in the keyword-driven test or the keyword sequence, to which you add a new keyword, Silk Central compares the preceding keywords with keyword combinations in all other keyword-driven tests and keyword sequences and recommends the keywords that most frequently follow the preceding combination of keywords.
 - If there are no keywords before the position in the keyword-driven test or the keyword sequence, but there are keywords after the current position, then Silk Central compares the succeeding keywords with keyword combinations in all other keyword-driven tests and keyword sequences and recommends the keywords that most frequently precede the succeeding combination of keywords.
- Additionally, Silk Central takes into account how similar the found keywords are. For example, if both the name and group of two keywords match, then Silk Central recommends these keywords higher in comparison to two keywords for which only the name matches.

Troubleshooting for Keyword-Driven Testing

Why do I get the error "No application configuration present" when trying to replay a keyword-driven test?

If you get this error, your keyword-driven test does not include a *Start application* keyword as the first keyword. Silk Test requires the *Start application* keyword to apply the application configuration of your project to the keyword-driven test. When you record a new keyword-driven test in Silk Test, Silk Test automatically adds the *Start application* keyword as the first keyword to the keyword-driven test.

With Silk Test 19.5 or later, parallel testing is enabled by default. When you upgrade from a Silk Test version prior to Silk Test 19.5 to a more recent Silk Test version, previously executing keyword-driven tests might no longer execute because each individual test is now expected to specify the application under test (AUT).

To workaroud this issue, disable parallel testing on the machine on which Silk Test is running by setting the environment variable `SILK_TEST_ENABLE_PARALLEL_TESTING` to false.

Business Process Tests

The intention of *business process testing (BPT)* in Silk Central is helping customers addressing the challenges of validating end-to-end business transactions. BPT is an extension to the existing manual testing capabilities of Silk Central, introducing new BPT-focused asset types to reduce complexity and to drive BPT-specific functionality.

Such end-to-end business transactions are a collection of related structured activities that can be conducted by one or more persons to achieve a particular business goal. These activities can either be related to a specific business unit or span across several business units. If they span across multiple business units, collaboration between these units is required to accomplish the business transaction. The activities that are involved in a business transaction can be an encapsulated part of other business transactions. Any individual activity can also contain another entire business transaction.

When mapping this approach into the testing space, one can imagine that the activities of former single tests can be connected in series to each other, where the results of one test might influence the execution

of the subsequent test. Additionally, each test in this sequence can be executed by a different tester representing a business unit, therefore collaboration and hand-off mechanisms are required.

BPT is an extension to the existing manual testing capabilities of Silk Central, introducing new BPT-focused asset types to reduce complexity and to drive BPT-specific functionality.

The following new asset types are introduced for BPT in Silk Central:

- A *business component* represents a basic component of an end-to-end business transaction and consists of the activities that can be conducted by a specific business unit. All tests in a business component can be performed by the same tester. A business component has the following characteristics:
 - It consists of manual test steps and shared steps.
 - It cannot contain other business components.
 - It cannot contain flows.
 - It cannot be assigned to a manual test or to a shared steps object.
 - A business component is executed by a member of a group of testers. The group needs to be assigned to the business component.
 - It can be maintained in the **Libraries** unit, which is accessible from the Silk Central menu under **Tests > Libraries**
- A *flow* represents a reusable sequence of activities that are included in multiple end-to end business transactions. A Flow can ideally be reused in multiple business process tests. A flow has the following characteristics:
 - It is a collection of business components in a defined sequence.
 - It cannot contain other flows.
 - It cannot contain manual steps or shared steps.
 - It cannot be assigned to a manual test or to a shared steps object.
 - It can be maintained in the **Libraries** unit, which is accessible from the Silk Central menu under **Tests > Libraries**
- A *business process test* represents an end-to end business transaction. A business process test has the following characteristics:
 - It is a collection of business components and flows in a defined sequence.
 - Each component is executed separately.
 - It can be maintained in the **Tests** unit.

Creating a Business Process Test

1. In the menu, click **Tests > Details View** .
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. Select the test type **Business Process Test** from the **Type** list.
6. In the **Planned time** field, type the expected amount of time for this business process test to execute.
7. Click **Finish**.
8. Continue by adding business components and flows to your new business process test.

Creating a Flow Out Of Business Components in a Business Process Test

Combine business components in a business process test to create a new flow.

1. In the menu, click **Tests > Details View** .
2. Select the business process test from the **Tests** tree.
3. Click the **Components** tab.
4. Select the business components that you want to include in the new flow.
5. Right-click on the selection.
6. Click **Create Flow**. The **Create Flow** dialog appears.
7. Type a name for the new flow into the **Name** field.
8. Select the location in the **Libraries** tree where the new flow should be located.
9. Click **OK**. Silk Central creates the new flow and replaces the selected business components in the business process test with the new flow. The business components in the new flow have the same behavior and the same assigned testers as in the business components in the original business process test.

Adding Components and Flows to a Business Process Test

Use the **Components** page to add business components and flows to a business process test.

1. In the menu, click **Tests > Details View** .
2. Select the business process test to which you want to add business components in the **Tests** tree.
3. Click the **Components** tab.
4. Select the business components and flows from the assignment tree.
 - To add a single business component or flow, click on the **Assign** arrow to the left of the business component or flow in the assignment tree.
 - To add all business components and flows in a folder or library, click on the **Assign** arrow to the left of the folder or library in the assignment tree.
 - To add multiple components and flows, select them in the assignment tree and click on the **Assign** arrow.



Tip: To assign multiple selected components and flows, you can also press **ENTER** or drag your selection into the grid.



Tip: You can filter the assignment tree based on libraries, folders, component names, and flow names.

Adding Business Components to a Flow

Use the **Components** page to add business components to a flow

1. In the menu, click **Tests > Libraries** .
2. Select the flow for which you want to add business components in the **Libraries** tree.
3. Click the **Components** tab.
4. Select the business components from the assignment tree.
 - To add a single business component, click on the **Assign** arrow to the left of the business component in the assignment tree.
 - To add all business components in a folder or library, click on the **Assign** arrow to the left of the folder or library in the assignment tree.
 - To add multiple business components, select the components in the assignment tree and click on the **Assign** arrow.



Tip: To assign multiple selected components, you can also press **ENTER** or drag your selection into the grid.



Tip: You can filter the assignment tree based on libraries, folders, component names, and flow names.

Assigning a Group of Testers to a Business Component

Use the **Components** page of a flow or a business process test to assign a group of testers to a business component.

Before you can assign a group of testers to a business component, ensure that the group is assigned to the project that contains the business process test or the flow.

1. Select the business process test or the flow that contains the business component to which you want to assign a group of testers.

- If the business component is included in a business process test, click **Tests > Details View**.
- If the business component is included in a flow, click **Tests > Libraries**.

2. Click the **Components** tab.

3. Right-click on the business component to which you want to assign a group of testers.



Note: You can only assign a group of testers to a component on the first hierarchy level, not to a sub-component of another component.

4. Click **Edit Assigned Testers**. The **Edit Assigned Testers** dialog box appears.

5. Select the group from the **Assigned Testers** list.



Note: This list displays only the groups that are assigned to the project that contains the business process test or the flow.

6. Click **OK**.



Note:

- For each Silk Central user, the **Manual Tests Assigned to Me** panel in the Silk Central start page shows the business components that are ready for execution.
- All members of the selected group are informed through an email that a business component is ready for execution.
- When the execution of a business component is finished, the members of the user group that is assigned to the next business component in the same flow are notified through an email.
- If no group is assigned to a business component, the settings from the **Deployment** tab are used.
- The `Executed by` property shows who is executing a business-process test. This property is visible in the **Manual Testing Window** and in the **Activities** page.



Note: You cannot assign groups of testers to a flow.

Specifying the Behavior in Case a Business Component Fails

Use the **Behavior on Failure** setting to specify what should happen to the execution of a business process test when the execution of a specific business component fails.

1. Select the business process test or the flow that contains the business components.

- To select a business process test, click **Tests > Details View**.
- To select a flow, click **Tests > Libraries**.

2. Select the business process test or the flow from the tree.

3. Click the **Components** tab.

4. Select the business components.

5. Right-click on the selection.

6. Click **Set Behavior on Failure**. The **Set Behavior on Failure** dialog appears.

7. Specify what the business process test should do if the business component fails:

- Select **Stop** from the list to stop the execution of the business process test.
- Select **Continue** from the list to continue the execution of the business process test.

Executing Business Process Tests

Once you have started your execution plan, you are ready to execute your business process tests.

1. Open the **Manual Testing** window.

To open the **Manual Testing** window, click **Home > My Dashboard** in the menu, and then click  (**Continue Manual Test**) in the panel **Manual Tests Assigned to Me**. If this panel does not display on your dashboard, you have to add it. For more information on the dashboard and on panels, see [Dashboard](#).

The **Manual Testing** window displays the Business Process Test in the same section where manual tests are shown. However, in contrary to manual tests, a Business Process Test contains an additional area where the currently executed business component is shown. The steps section represents the steps for the current business component. For additional information on the window, see [Manual Testing Window](#).

2. Execute the steps in the steps section.

3. *Optional*: Specify values for any output parameters of the current business component.

All parameters for the current component and their value are listed after the component description.

Any input parameters that are using the `${name}` syntax are replaced on execution with their current values.

The output parameters are listed below the input parameters with an input field to specify a value. If no value is specified, the user is warned when finishing the component. However, it is not mandatory to set a value to an output parameter when finishing a component.

For additional information on how to specify and map parameters within components and flows, see the following topics:

- [Specifying Parameters for Business Components](#)
- [Specifying Parameters for Flows](#)
- [Mapping Parameters for Business Process Testing](#)

4. When you have executed all test steps for the business component, click **Finish Component**. The component is marked as finished and the execution of the next available component of the Business Process Test is started.

 **Note:** You can set the status of the business component to finished at any time, however, once the component is finished, you cannot change the status of the component anymore.

5. Continue until all components that are assigned to you are finished.

 **Note:** You can also execute business process tests from the current run tab of the execution plan or from the **Activities** page.

Generating Business Components from Manual Tests

Generate new business components out of manual tests to reuse the steps in the manual tests for business process testing.

1. In the menu, click **Tests > Details View**.

2. Select one or more manual tests in the **Tests** tree.

3. Right-click your selection and select **Generate Business Component**.

4. Specify how you want to handle the steps in the new business components.

- Click **Copy the steps from the manual tests** to copy the steps of each selected manual test into a corresponding new business component. Future changes to the steps of the manual test will not affect the steps of the corresponding business component and the other way round.

- Click **Share and reuse the steps of the manual tests** to copy the steps of each selected manual test into a corresponding new shared steps asset. Each shared steps asset is then used by both the manual test and the new business component. Future changes to a shared steps asset will affect both the manual test and the business component.
5. Choose the location where you want to place the new business components in the **Libraries** tree.
 6. Click **OK**. Silk Central creates a new business component for each selected manual test.

Example

For example, suppose you have selected the manual tests *Test 1* and *Test 2*, and you want to create two new business components out of these manual tests.

- If you choose to generate business components out of these tests by copying the steps from the manual tests, Silk Central generates two new components with the names *Test 1* and *Test 2*. The new component *Test 1* has the same steps as the manual test *Test 1*, and the new component *Test 2* has the same steps as the manual test *Test 2*.
- If you choose to generate business components and to reuse the steps of the manual tests, Silk Central performs the following actions:
 - It generates two new shared steps with the names *Test 1* and *Test 2*.
 - It replaces the steps in the manual test *Test 1* with the new shared step *Test 1*.
 - It replaces the steps in the manual test *Test 2* with the new shared step *Test 2*.
 - It generates two new business components with the names *Test 1* and *Test 2*. The new business component *Test 1* has the new shared step *Test 1* and the new business component *Test 2* has the new shared step *Test 2*.

Specifying Parameters for Business Components

You can specify input and output parameters to generate a data flow between business process tests, business components, and flows. For example, you might want to pass user credentials from a test into its business components or flows, or you might want to pass an *Order ID* parameter that is generated from a business component to be further processed in subsequent business components or flows.

As their names imply, input parameters provide data to the testers of a specific business component, while output parameters represent the data that is produced by the testers of the business component with the intention to pass information on to the testers of subsequent business components.

To specify new parameters for a business component or to edit existing parameters of a business component:

1. In the menu, click **Tests > Libraries**.
2. Select the business component for which you want to specify the parameters from the **Library** tree.
3. Click the **Steps** tab. All parameters that are currently specified for the business component are displayed next to the **Parameters...** button.
 - Input parameters are displayed as $\${name}$.
 - Output parameters are displayed as $\${name}$.
4. Click **Parameters...**. The **Parameters** dialog opens.
5. Specify the new parameters.

For each new parameter, perform the following actions.

- a) Type a name for the new parameter. Parameter names must be unique in the context of the business component.
- b) To specify the new parameter as an output parameter, check the **Output** check box.
- c) Click **Add**.

6. When you have specified all required parameters, click **OK** to close the **Parameters** dialog.

You can reference the specified input parameters in the **Action Description** or **Expected Results** section of the manual test steps by using the syntax $\${name}$, for example `user: $\${username}$` . Output parameters are set during the actual execution of the business process test.

Specifying Parameters for Flows

You can specify input and output parameters to generate a data flow between business process tests, business components, and flows. For example, you might want to pass user credentials from a test into its business components or flows, or you might want to pass an *Order ID* parameter that is generated from a business component to be further processed in subsequent business components or flows.

As their names imply, input parameters provide data to the testers of one or more specific business components in a flow, while output parameters represent the data that is produced by the testers of one or several business components with the intention to pass information on to the testers of subsequent business components.

To specify new parameters for a flow or to edit existing parameters of a flow:

1. In the menu, click **Tests > Libraries**.
2. Select the flow for which you want to specify the parameters from the **Libraries** tree.
3. Click the **Components** tab. All parameters that are currently specified for the flow are displayed next to the **Parameters...** button.
 - Input parameters are displayed as $\${name}$.
 - Output parameters are displayed as $\${name}$.
4. Click **Parameters....** The **Parameters** dialog opens.
5. Specify the new parameters.

For each new parameter, perform the following actions.

 - a) Type a name for the new parameter. Parameter names must be unique in the context of the flow.
 - b) To specify the new parameter as an output parameter, check the **Output** check box.
 - c) Click **Add**.
6. When you have specified all required parameters, click **OK** to close the **Parameters** dialog.

Mapping Parameters for Business Process Testing

Use the **Components** tab to map the parameters of a business component in the context of a business process test or a flow to one of the following:

- An input parameter of the selected business process test or flow.
- An output parameter of a previous business component in the selected business process test or flow.
- A test parameter of the business process test. This is not applicable to business components in the context of a flow. For additional information, see [Test Parameters](#).
- A static value.



Note: Parameters that match by name, are mapped automatically. For example, the output parameter $\${name}$ is automatically mapped to the input parameter $\${name}$ of any following component in the business process test or flow.

1. Select the business process test or the flow.
 - To select a business process test, click **Tests > Details View** and select the business process test from the **Tests** tree.
 - To select a flow, click **Tests > Libraries** and select the flow from the **Libraries** tree.
2. Click the **Components** tab. The **Parameters** column displays all parameters that have been specified for the assigned components.

3. Click on the parameter that you want to map.

4. Specify the mapping for the parameter.

- To map the parameter to another parameter, select the parameter to which you want to map from the list.
- To set a static value for an input parameter, type the value into the text field and press **Enter**.



Caution: When you rename or re-order the parameters of a component, the parameter mapping may become invalid and you will have to remap the parameters.

Example

Assume you have a sequence of business components to order, procure, and ship products.

| | Actions | Component | Parameters | | |
|---|---------|---------------|-----------------|-----------------------------|------------------------------|
| 1 | ✗ Q? | Order Product | $\$(ProductID)$ | $\$(OrderID) \rightarrow$ | |
| 2 | ✗ Q? | Procurement | $\$(OrderID)$ | $\$(InvoiceID) \rightarrow$ | |
| 3 | ✗ Q? | Shipping | $\$(OrderID)$ | $\$(InvoiceID)$ | $\$(TrackingID) \rightarrow$ |

A product is ordered in the first business component and this results in an `OrderID` which is then passed on to the subsequent business components.

Silk Central automatically maps the `OrderID` output parameter of the first business component to the `OrderID` input parameters of the subsequent business components because the parameters have the same name.

If you have multiple business components with an output parameter of the same name, the input parameters with the same name of subsequent components will be mapped to the output parameter of the most recently preceding business component in the sequence.

Assume you have the sequence that is displayed in the following image.

| | Actions | Component | Parameters | | |
|---|---------|---------------|-----------------|-----------------------------|------------------------------|
| 1 | ✗ Q? | Order Product | $\$(ProductID)$ | $\$(OrderID) \rightarrow$ | |
| 2 | ✗ Q? | Order Product | $\$(ProductID)$ | $\$(OrderID) \rightarrow$ | |
| 3 | ✗ Q? | Procurement | $\$(OrderID)$ | $\$(InvoiceID) \rightarrow$ | |
| 4 | ✗ Q? | Shipping | $\$(OrderID)$ | $\$(InvoiceID)$ | $\$(TrackingID) \rightarrow$ |
| 5 | ✗ Q? | Procurement | $\$(OrderID)$ | $\$(InvoiceID) \rightarrow$ | |
| 6 | ✗ Q? | Shipping | $\$(OrderID)$ | $\$(InvoiceID)$ | $\$(TrackingID) \rightarrow$ |

The output parameter of the first product order in the sequence cannot be accessed in any subsequent business component as it is overwritten by the output parameter of the second product order.

To prevent this from happening, you can write the value of output parameters into temporary variables by clicking into the output parameter field and by typing `{<Variable Name>}`. You can then use the temporary variable as an input parameter to subsequent business components.

The following image shows how this works by writing the `OrderID` output parameters of the two subsequent *Order Product* business components to the temporary variables *OrderID A* and *OrderID B*.

| | Actions | Component | Parameters | | |
|---|---------|---------------|-----------------|-----------------------------|------------------------------|
| 1 | | Order Product | $\$(ProductID)$ | $\$(OrderID A)$ | |
| 2 | | Order Product | $\$(ProductID)$ | $\$(OrderID B)$ | |
| 3 | | Procurement | $\$(OrderID A)$ | $\$(InvoiceID) \rightarrow$ | |
| 4 | | Shipping | $\$(OrderID A)$ | $\$(InvoiceID)$ | $\$(TrackingID) \rightarrow$ |
| 5 | | Procurement | $\$(OrderID B)$ | $\$(InvoiceID) \rightarrow$ | |
| 6 | | Shipping | $\$(OrderID B)$ | $\$(InvoiceID)$ | $\$(TrackingID) \rightarrow$ |

Importing Business Process Tests from an Excel File

Import business process tests from a Microsoft Excel file to add them to the **Tests** area.

1. In the menu, click **Tests > Details View**.
 2. In the **Tests** tree, right-click the folder or container that you want to use as the parent node of the imported test and choose **Import from Excel File**. The **Import Tests from Excel File** dialog box opens.
 3. Click **Business Process Tests**.
 4. Select the mapping file from the **Choose Mapping File** list box.
The default mapping file is `TestPropertiesDefaultMapping.xml`. For information about creating customized mapping files, see *Editing the Test Mapping File*.
 5. *Optional:* Click to edit the mapping file. Click to delete the mapping file.
- Note:** These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.
6. Select the **Library** in which the business components for the business process tests should be created. Business components that already exist in the selected library will be reused.
 7. Either type the fully qualified path of the test file into the **Test File** field or click **Choose File** to search for the test file.
 8. Click **OK**.

The business process tests are imported directly under the selected node without creating a common root folder.

Unified Functional Testing (UFT) Tests

Unified Functional Testing (UFT) software, formerly known as HP QuickTest Professional (QTP), provides functional and regression test automation for software applications and environments.

Creating or Editing Unified Functional Testing (UFT) Tests

To execute UFT tests, the Unified Functional Testing software must be installed on the same computer on which Silk Central execution servers are installed.

To create or edit a UFT test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click (**New Child Test**) on the toolbar, or click (**Edit**).
The **New Test/Edit Test** dialog box appears.
4. Type a name and description for the test.

 **Note:** Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Unified Functional Testing Test** from the **Type** list box. Click **Next**.

The **Unified Functional Testing Test Properties** dialog box appears.

6. In the **Test** field, select the source control location of the UFT test to be executed.

 **Tip:** You can select the test (.tsp) itself, or the folder where the test is located.

7. *Optional:* Check the **Validate UFT add-ins** option to validate all specified add-ins of the selected UFT test.
8. *Optional:* Check the **Show UFT application** option to display the application under test during test execution.
9. *Optional:* Check the **Log debug messages** option to log debug messages in the test run results.
10. *Optional:* Use the **Inherit from parent** check boxes to inherit the settings from the parent UFT test.
11. Click **Finish**.

 **Note:** Parameters are passed to the UFT test by defining parameters with the same name as the test parameters that are defined in the UFT test itself.

The original UFT result files, `run_results.xml` and `run_results.html`, are available in the **Files** tab of the test run results.

JUnit Tests

JUnit is a unit testing framework for the Java programming language. Use this test type for JUnit, Silk4J, Selenium, and UFT Developer tests.

Creating or Editing JUnit Tests

To create or edit a JUnit test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.

 **Note:** Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **JUnit Test** from the **Type** list box. Click **Next**.

The **JUnit Test Properties** dialog box appears.

6. Specify a valid Java **Classpath** to use on the execution server.

We recommend to use a relative classpath. The relative classpath is then expanded to the full classpath on the execution server. By using a relative classpath, changes on the location of the source control profile do not require additional changes to the classpath.

The relative classpath must point to the root node of the test container containing the JUnit test, for example `JUnit_tests`. The relative classpath on the execution server is then expanded to include the working folder of the source control profile, for example `C:\temp`, and the test file names, for example `tests.jar`. For example, if your tests are packaged in a jar file called `tests.jar` and additional jar files that are needed for the execution are stored in a sub folder `lib`, your classpath would be `./tests.jar;./lib/*`.

You can also use a fully qualified classpath. The fully qualified classpath must point to the archive or folder in which the test classes reside.

7. *Optional:* In the **Test class** field, type the fully qualified name of the JUnit test class. If the field is left blank, all tests in the classpath will be executed.
8. *Optional:* In the **Test method** field, type the name of the appropriate test method.
The method must be available in the test class. If the **Test method** field is left blank, all tests that are included in the suite will be executed.
9. *Optional:* Set the **Java home directory** to the installation path of the Java Runtime Environment (JRE).
The path must be valid on the execution server on which the test runs.



Note: JUnit tests can be executed in the latest version of JRE 1.8. If you use an older JRE version, messages containing `java.lang.UnsupportedClassVersionError` or `Unrecognized option: -javaagent` will display in the **Messages** tab.

10. *Optional:* In the **JVM options** field, you can specify the command-line options. You can specify multiple options, but you have to type them in the right order. For example, to use the client VM and set the maximum size of the heap to 512MB, type `-client -Xmx512m`. You can also use environment variables, for example `-Xmx${MY_MEMORY}`.
11. Check the **Record external AUT coverage only** check box to get only the code coverage for the application under test that is defined for the execution plan in the **Code Analysis Settings** portion of **Silk Central > Execution Planning > Code Analysis** .
If the check box is not checked, then the code coverage from the virtual machine that executes the test is recorded too.
12. *Optional:* In the **Test coverage path** field, type the JAR libraries or the specific class files to monitor for code coverage information. Note that the coverage path of the JUnit test is specified here, not the path of the application under test.

We recommend using the relative coverage path from the test container root node, which is then expanded on the execution server. You can also use a fully qualified path. Use semicolons to separate multiple jar files, as the following examples show:

- `tests.jar;C:\tests\testutils.jar`
- `${testenv}\tests.jar;${testenv}\testutils.jar`



Note: The coverage path setting is disregarded if the **Record external AUT Coverage** check box is checked.

13. Click **Finish**.



Note: Parameters are passed to the Java process as system properties, for example `-Dhost_under_test=10.5.2.133`. Use the `System.getProperty()` method to access the system properties. For example, to access the previously passed `host_under_test`, use `System.getProperty("host_under_test");`

Accessing Silk Central Parameters through Java System Properties

Any JUnit test class can access a test parameter of the underlying test as a Java system property; the launcher passes these parameters to the executing machine using the "-D" VM argument.

In addition to the customer-defined parameters, you can always call the following Java system properties from a JUnit test:

| Parameter | Description |
|---------------------------------|---|
| <code>#sctm_execdef_name</code> | The name of the execution plan that is currently executed. If the execution was triggered from the Tests area, the parameter has the value <code>Unassigned Tests</code> . |

| Parameter | Description |
|---|---|
| #sctm_execdef_id | The numeric identifier (database key) of the execution plan that is currently executed. |
| #sctm_product | The name of the product as defined in the executed test container. |
| #sctm_version | The name of the version to which the results of the execution are associated. |
| #sctm_build | The name of the version to which the results of the execution are associated. |
| #sctm_keywords | A comma separated list which contains the keywords that are defined for this execution plan. |
| #sctm_regular_execdef_run_id | The numeric identifier (database key) of the regular execution plan run. For setup and cleanup tests this property contains the run ID of the regular run. |
| #sctm_test_results_dir | The path of the directory where the result files of the test reside. |
| #sctm_test_name | The name of the Silk Central test being executed. |
| #sctm_test_id | The numeric identifier (database key) of the test being executed. |
| #sctm_source_root_dir | The local path of the test container's root directory, where all files under version-control reside. |
| #external_id | The unique identifier of a test case in a test package to be executed. You can use this as a filter if only specific test cases in a test package should be executed. |
| #is_test_package_root | If <code>true</code> , the test being executed is the root node of the test package (default). If <code>false</code> , only specific test cases in a test package specified by <code>#external_id</code> are executed |
| #sctm_data_driven_parent_test_name | The name of the parent test, if the executed test is a data driven row. |
| #sctm_data_driven_parent_test_id | The numeric identifier (database key) of the parent test, if the executed test is a data driven row. |

MSTest Tests

MSTest is a command line utility from Microsoft that executes unit tests created in Visual Studio. Use this test type for MSTest, Silk4NET, and UFT Developer tests.

Creating or Editing MSTest Tests



Note: The MSTest plugin supports the test type unit tests only. Results of other test types are ignored.

To create or edit an MSTest test:

1. In the menu, click **Tests > Details View** .
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **MSTest Test** from the **Type** list box. Click **Next**.

The **MSTest Test Properties** dialog box appears.

6. You can run a test container, a test list, an ordered test, a test class, or a single test method:

- To run a test container, click **Browse** next to the **Test file** field and select a `.dll` file. You can constrain the run by typing in just a **Test class** or both a **Test class** and a **Test method**.
- To run a test list, click **Browse** next to the **Test file** field and select a `.vsmdi` file (Visual Studio Test Meta Data). Type a name in the **Test list name** field. You can constrain the run by typing in both a **Test class** and a **Test method**.
- To run an ordered test, click **Browse** next to the **Test file** field and select a `.orderedtest` file.



Important: To run ordered tests, leave all fields except the **Test file** field blank.

7. Click **Finish**.



Note: To run MSTest on an execution server, a Visual Studio distribution or the Visual Studio Test Agent have to be installed. Add the folder that contains `MSTest.exe` to your path variable and restart the execution server. The default path is: `C:\Program Files (x86)\Microsoft Visual Studio <version>\Common7\IDE`. Currently MSTest distributed with Microsoft Visual Studio/Visual Studio Test Agent 2015 is supported.

NUnit Tests

NUnit is a unit testing framework for the Microsoft .NET programming language.

Creating or Editing NUnit Tests



Note: We recommend that you add the `.\bin` folder of your NUnit installation to the system path. Click **Start > Control Panel > System > Advanced > Environment Variables** to add a path like `C:\Program Files\NUnit 2.2\bin` to the system environment variable `PATH`.

To create or edit an NUnit test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **NUnit Test** from the **Type** list. Click **Next**.

The **NUnit Test Properties** dialog box appears.

6. Click **Browse** to locate and select the NUnit assembly from which you want to pull a test.

7. Type the working directory in the **NUnit Directory** field.

For NUnit 2, this directory is the local path to the file `nunit-console.exe`, such as `C:\Program Files\NUnit 2.2\bin`. For NUnit 3, this directory is the full path to the file `nunit3-console.exe`, such as `C:\Program Files (x86)\NUnit.org\nunit-console\nunit3-console.exe`.

8. In the **NUnit Options** field, type one or more NUnit console command-line options to specify how NUnit tests are specified.

For example, to define that the `Flex4TestApp` NUnit test, located in `SilkTest.Ntf.Test.Flex`, should be executed, set the run option as follows:

For NUnit 2:

```
/run:SilkTest.Ntf.Test.Flex.Flex4TestApp
```

For NUnit 3:

```
/test:SilkTest.Ntf.Test.Flex.Flex4TestApp
```



Note: When you add multiple options, you have to separate the options by writing one option in each line in the field.

9. Click **Finish**.

ProcessExecutor Tests

The `ProcessExecutor` test type can be used to execute any command line.

Creating or Editing ProcessExecutor Tests

To create or edit a `ProcessExecutor` test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **ProcessExecutor Test** from the **Type** list box. Click **Next**. The **ProcessExecutor Test Properties** dialog box appears.
6. In the **Executable Name** field, type the fully qualified name of the executable.
7. In the **Argument List** field, type all arguments of the process executor test method.



Note: Multiple arguments must be on separate lines in the field.

8. Set the **Working Folder**. This is the folder where the executable is executed.
9. If the executed command produces JUnit test result files, for example an Ant or Maven JUnit execution, you can specify a folder pattern that is scanned for JUnit result xml files. The folder location is relative to the working directory, for example `test-results`. The root folder where the search starts is the **Working Folder**, in which all JUnit output files where the path contains the entered pattern are retrieved.

During execution of the executable the following two environment variables can be used:

| Environment Variable | Description |
|--------------------------------|--|
| SCTM_EXEC_RESULTSFOLDER | All files in this folder are saved as result files in Silk Central. If an <code>output.xml</code> file is created during execution in this directory, it is processed by Silk Central. |
| SCTM_EXEC_SOURCESFOLDER | This is the folder where all source files used during the execution are located. |



Note: To define these environment variables as arguments for a test, use the predefined parameters that are available for all test types, instead of using the specific environment variable that is only

available for ProcessExecutor tests. For example, use #sctm_test_results_dir instead of SCTM_EXEC_RESULTS_FOLDER.

Examples

To configure a ProcessExecutor test that runs the Windows Script Host from the command line with the /c switch and two parameters, specify the following:

- /c - A command that specifies that the command line should terminate after execution.
- cscript - The Windows Script Host.
- parareadwrite.js - The jscript file.

To run the test on a Linux server, you can use the following Python script:

```
#!/usr/bin/env python
#
import sys
import os
print 'command and arguments: ' + str(sys.argv)
print 'SCTM_EXEC_RESULTS_FOLDER: ' +
str(os.environ.get('SCTM_EXEC_RESULTS_FOLDER'))
```

Instead of using the environment variable SCTM_EXEC_RESULTS_FOLDER, you can set the predefined parameter #sctm_test_results_dir as an argument in Silk Central. The argument will be replaced with the corresponding environment variable when the script is executed.

Silk Performer .NET Explorer Tests

The Silk Performer .NET Explorer test type is used to execute .NET Explorer scripts or a specific test case of a script.

Creating or Editing Silk Performer .NET Explorer Tests

To create or edit a .NET Explorer test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Performer .NET Explorer Test** from the **Type** list box. Click **Next**.

The **.NET Explorer Test Properties** dialog box opens.

6. Browse to and select the .NET Explorer script to apply to the test.

The script is a .nef file.

7. Enter the path to the NetExplorer.exe executable on the execution server.

For example C:\Program Files\MyCustomSPFolder\DotNET Explorer\NetExplorer.exe. You can also define an environment variable on your execution servers and refer it, for example \${SP_HOME}\NetExplorer.exe.

8. In the **Test case** field, type the name of the .NET Explorer test case to execute. If this field is left blank, all test cases within the script are executed. It is not possible to specify multiple test cases.



Note: The test cases `InitTestCase` and `EndTestCase` are always executed.

9. Click **Finish**.

Silk Performer Tests

Creating or Editing Silk Performer Tests

To create or edit a Silk Performer test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Performer Test** from the **Type** list box. Click **Next**.

The **Silk Performer Test Properties - Select Project** dialog box appears.

6. Click **Browse** to select a Silk Performer project that has been saved to your file system and click **Next**.
7. Select one of the workload profiles that has been defined for the project from the **Workload** list.
8. Click **Finish** to create the test case.

Silk Test Classic Tests

Silk Central's Silk Test Classic interface offers a reliable means of automating Silk Test Classic users. Each test case of a Silk Test Classic script executes within its own test execution and produces its own results.

The Silk Test Classic Plan test type is used for Silk Test Classic test plans. A test plan usually is a hierarchically-structured document that describes the test requirements and contains the statements, 4Test scripts, and test cases that implement the test requirements.

When you observe running Silk Test Classic executions on the **Activities** page, the currently running execution plan offers a hyperlink that opens a **Details View**. This view allows you to closely monitor the state of the currently running execution plan. For Silk Test Classic test executions, the center component of this view consists of two parts: The upper part shows general information about the test, script, test case, and test data. The lower part shows all output messages generated by Silk Test Classic, along with their severity.

In previous versions of Silk Central, Silk Test Classic invocation was implemented through a command-line interface. The new interface works using interprocess communication. You can specify whether or not Silk Central's Silk Test Classic interface should be used by configuring test-container settings.

For all Silk Test Classic test parameters that are not defined through Silk Test Classic test properties in the Silk Central GUI, the Silk Test Classic default settings are used, for example from `partner.ini`.

Creating or Editing Silk Test Classic Tests

To create or edit a Silk Test Classic test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.

3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Test Classic Test** from the **Type** list box. To import multiple Silk Test Classic test cases at the same time, select **Silk Test Classic Multi-testcase import**. Click **Next**.

The **Test Properties - Select Test Script** dialog box appears.

6. Click **Browse** and select the test script file from either the defined Silk Test Classic project or the source control directory.

The test script, with file extension `.t` or `.g.t`, is defined relative to the test container's root node in the source control profile. For example `<Silk Test Classic project>/MyScript.t`.

7. Click **Next**.

The **Test Properties - Select Testcase** dialog box appears.



Note: If the Silk Test Classic script is a data-driven `.g.t` file, for example `SilkTestScript1.g.t`, then data sources are completely controlled within the script file and not through the data-driven properties of Silk Central. The **Data-driven** check box is checked by default when you use a data-driven script file. For more information about data-driven Silk Test Classic tests, refer to the Silk Test Classic documentation.

8. Select a test case from the available test cases in the defined script file or specify a custom test case. When the custom test case field is already populated, the Silk Test Classic test was automatically created, using the export functionality within Silk Test Classic. When you use the custom field to specify the test case, you can terminate the test case name with a parenthesis `()`. Inside the parenthesis, you can specify test data, including parameters. Specifying data in the custom field will override the values of the **Test data** property.

9. *Optional:* Type additional execution arguments into the **Test data** field.

These arguments are processed by Silk Test Classic during the execution of the test. If several arguments are passed to Silk Test Classic, they have to be separated by a comma `,`. If a `String` argument is passed to Silk Test Classic, the argument must be set in quotation marks `"`. When test data is more complex, we recommend that you use parameters in the test data, for example `${ParameterName}`. Parameters are replaced automatically within test executions.



Note: This field has a maximum length of 2000 characters.

10. Define whether the test should provide a TrueLog.



Tip: When running your tests with the Open Agent, this setting must be turned on both in Silk Central and in Silk Test, otherwise no TrueLog will be provided. Default execution mode for data-driven tests is plan-based. When you use script-based execution mode for a data driven test, change the `DataDrivenScriptMode` setting in the Silk Test Classic element of `SccExecServerBootConf.xml`. For more information, refer to the Silk Test Classic documentation.

11. If required, specify an option set file. By default, Silk Central closes all open Silk Test Classic option set files. To specify an option set file, specify the file name relative to the test container's root node in the source control profile.

12. Click **Finish** to create the Silk Test Classic test.



Note: If you possess Silk Test Classic test cases that require more than one hour to complete, adjust the time-out settings for Silk Central. Otherwise, Silk Central assumes an error has occurred

and terminates the execution. For details about time-out settings, see the *Administration* topics in this Help.

Creating or Editing Silk Test Classic Plan Tests

To create or edit a Silk Test Classic plan test:

1. In the menu, click **Tests > Details View** .
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Test Classic Plan** from the **Type** list box. Click **Next**.

The **Plan Properties** dialog box appears.

6. In the **Plan file** field, type the fully qualified name of the test plan file (.pln) to be executed.
Click **Browse** to browse for the file.
7. In the **Project file** field, type the name of the Silk Test Classic project containing the file and environmental settings.
Click **Browse** to browse for the project file.
8. In the **Option set** field, type the fully qualified name of the option set file containing environmental settings.
Click **Browse** to browse for the option set file.
9. In the **Data file for attributes and queries** field, type the default path of the test plan initialization file.
Click **Browse** to browse for the test plan initialization file.
10. In the **Test plan query name** field, type the fully qualified name of the saved test plan query.
11. Click **Finish**.

To maintain the structure of your Silk Test Classic plan test in Silk Central, execute a trial run. Once the trial run has completed successfully, the `output.xml` file is added to the result files of your execution plan run. This allows you to convert your test to a test package, giving you full control over the test's structure.

Silk Test Workbench Tests

Silk Test Workbench is an automated testing tool that accelerates the functional testing of complex applications. Silk Test Workbench provides support for testing applications developed in a wide variety of development tools including Java, .NET, browser-based web applications, and COM components, including both ActiveX controls and automation objects.

Prerequisites to Run STW.EXE with Silk Central

To run STW.EXE with a Silk Central execution server and to successfully connect to a Silk Test Workbench database, the following prerequisites are required:

1. Run the Silk Central execution server as a process, and not as a service.
2. Use database authentication to connect to the Silk Test Workbench database. Silk Test Workbench can use Windows authentication or database authentication when connecting to a Silk Test Workbench SQL Server database.
3. Configure the database server instance to use a static port.

If these prerequisites are not met, Silk Test Workbench might return the following error when trying to run STW.EXE with a Silk Central execution server:

Error 3: DSN, username, or password incorrect.

Creating or Editing Silk Test Workbench Tests

To create or edit a Silk Test Workbench test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Test Workbench Test** from the **Type** list box. Click **Next**.

The **Silk Test Workbench Test Properties - Select Tests** dialog box appears.

6. Select one or more scripts from the **Silk Test Workbench Test** list box.
7. In the **Playback Options** field, type in the name of a predefined Silk Test Workbench playback option or leave the default value of `System Defaults`.



Note: A playback option must be set for a Silk Test Workbench test. The delete option on a Silk Test Workbench test will reset the playback option to the default value of `System Defaults`.

8. Click **Finish**.

Opening Silk Test Workbench in the Context of the Result File

Click the `result.stwx` file to open Silk Test Workbench in context of the result file `result.stwx`.

If Silk Test Workbench uses an Oracle database, ensure that the file `tnsnames.ora` exists on the machine on which you want to open Silk Test Workbench, and that the file includes an entry with a `net_service_name` that is the same as the `SID` that is configured in the Silk Test Workbench source control. For example:

```
myOrclSid=
(DESCRIPTION=
(AADDRESS=... )
(CONNECT_DATA=
(SID=myOrclSid)))
```

1. In the menu, click **Execution Planning**.
2. Select your execution plan.
3. Click the **Runs** tab.
4. In the **Test Runs** grid at the bottom, click the **Run ID** of the test for which you want to see details. The **Test Run Results** dialog box opens.
5. Click the **Files** tab.
6. Click the `result.stwx` file to open Silk Test Workbench in context of the result file.



Note: Silk Test Workbench must be installed on the machine on which you are trying to open the file.

Viewing the Execution Result Files for a Visual Test

1. In the menu, click **Execution Planning**.
2. Select your execution plan.

3. Click the **Runs** tab.
4. In the **Last Executions** area, click the **Run ID** of the relevant execution plan to view test-execution results.
5. The **Actions** column displays all icons and links for Silk Test Workbench:
 - `detail.htm` - Silk Test Workbench detailed steps file.
 - `result.stwx` - Click to open the result in Silk Test Workbench .
 - `error.png` - Click to go to a screen shot of the last error.

Test Parameters Page - Silk Test Workbench

Tests > Details View > <Test Element> > Data Set

When passing parameters from a Silk Central test to a Silk Test Workbench test, these parameters must be defined as input parameters in the Silk Test Workbench script and they must have the same name in Silk Central as in the script. The following data types should be used:

| Silk Central | Silk Test Workbench |
|--------------|----------------------|
| String | Text |
| Number | Number (Long) |
| Number | Number (Long Long) |
| Number | Enumeration |
| Float | Number (Double) |
| Boolean | Boolean (True/False) |

Predefined parameter names in Silk Central start with a hash while Silk Test Workbench does not allow parameter names starting with a hash. To pass a predefined parameter from Silk Central to a Silk Test Workbench test, you have to create an additional parameter without the hash in the name in Silk Central and map the predefined parameter to the new parameter.

For example, to pass the predefined parameter `#sctm_test_results_dir` to a Silk Test Workbench script, you have to apply the following workaround:

1. Specify an input parameter in the Silk Test Workbench script with the name `resultsdir`.
2. Specify a parameter in Silk Central with the name `resultsdir` and the value `#{sctm_test_results_dir}`.

Silk Central will then replace the placeholder with the actual directory and pass it to the Silk Test Workbench script.

Silk TestPartner Tests

TestPartner is a Micro Focus product that tests Web- and Microsoft Windows-based applications that use Microsoft technologies.

Creating or Editing Silk TestPartner Tests

To create or edit a TestPartner test:

1. In the menu, click **Tests > Details View** .
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk TestPartner Test** from the **Type** list box. Click **Next**.

The **Test Properties - Select Test Script** dialog box appears.

6. Click **Browse**.

The **Choose File** dialog box opens. The assets available in this dialog box are based on the **Project Path** defined in the associated TestPartner source control profile.

7. Select a script from the list and click **OK**.



Note: To add multiple TestPartner tests, see *Adding Multiple TestPartner Tests*.

8. In the **Playback Options** field, type in the name of a predefined TestPartner playback option or leave the default value of `System Defaults`.

9. Click **Finish**.

Adding Multiple TestPartner Tests

To add multiple TestPartner tests, you must first follow the steps described in *Creating Silk TestPartner Tests*.

To add multiple TestPartner tests:

1. On the **New Test** dialog box, select **TestPartner Test** from the **Type** list box.
2. Click **Next**. The **Test Properties - Select Test Script** dialog box opens.
3. Click **Browse**.

The **Choose File** dialog box opens. The assets available in this dialog box are based on the **Project Path** defined in the associated TestPartner source control profile.

4. In the **Playback Options** field, type in the name of a predefined TestPartner playback option or leave the default value of `System Defaults`.



Note: The import gives the defined playback option to all imported scripts. If you need to change the playback option for any scripts, edit the test after the import.

5. Click **Next**.

The **Test Properties - Select Scripts** dialog box opens.

6. In the **TestPartner Scripts** field, select the scripts to import by clicking **Ctrl+Click**.
7. Click **Finish**.

Windows Script Host Tests

Windows Script Host (WSH) is part of the Windows platform and creates an environment for hosting scripts. When a script is to be run at the execution server, WSH plays the role of host. It makes objects and services available for the script and provides a set of guidelines within which the script is executed. Among other things, WSH manages security and invokes the appropriate script engine.

Creating or Editing Windows Script Host (WSH) Tests

To create or edit a WSH test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.

 **Note:** Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Windows Script Host Test** from the **Type** list box. Click **Next**.

The **Windows Scripting Test Properties** dialog box appears.

6. Click **Browse** and select a Windows scripting test script.

7. Specify the location of any required additional parameters in the **Switches** field.

 **Note:** You may add other switches to be passed to the script. For more details on the switches that can be used, see [WSH Test Properties](#) and refer to the MS Scripting Host documentation.

8. Click **Finish**.

Supported Script Languages

WSH is language-independent for WSH-compliant scripting engines. Natively, the Windows platform supports Visual Basic Scripts, with file extension `.vbs`, and scripts written in the Java Script language, with file extension `.js`.

For other scripting languages, a dedicated script interpreter must be installed on the execution server. For example, if you install a Perl interpreter on an execution server, this will register a Perl scripting engine at the WSH environment for the extension `.pls`. Whenever a file with extension `.pls` is passed to the WSH tool, with the executable `cscript.exe`, it will invoke the appropriate interpreter because of the file extension. So the client of WSH, in this case the Silk Central Execution Server, does not need to know about the installation of the Perl interpreter.

 **Note:** After installing a script interpreter, for example Active Perl, try to execute a script locally on the execution server by calling the WSH command line tool with a sample script before executing the script in Silk Central. To do so, open a command shell on the execution server and type `cscript <script>`, where `<script>` is the path to a script of your choice that is available on your execution server. This is exactly what Silk Central will call when executing a WSH test on an execution server. If the script is executed, then the scripting engine has been registered successfully.

The following scripting languages are WSH compatible:

| Scripting Language | File Extension |
|--------------------|--------------------------------------|
| Perl | <code>.pls</code> |
| Python | <code>.PY</code> , <code>.PYW</code> |
| REXX | <code>.REXX</code> |
| TCL | <code>.tcl</code> |

WSH Test Properties

| Property | Description |
|----------|---|
| Script | You can define any file where a script engine is registered for the script language the file contains. Script files under source control are deployed automatically to execution servers, comparable to test sources for other test types. |
| Switches | You can enter and pass the following settings to <code>cscript.exe</code> during the execution of the test: //B Batch mode suppresses all non-command-line console UI requests from the script. We recommended that you use this option to prevent a script from waiting for user input during unattended executions at the execution server. |

| Property | Description |
|-------------------------|--|
| <code>//U</code> | We recommend to that you use unicode for redirected I/O from the console. |
| <code>//T:nn</code> | Time-out, in seconds. The maximum time the script can run, by default = no limit. This option is used to prevent excessive execution of scripts. It sets a timer. When execution time exceeds the specified value, Cscript interrupts the script engine using the <code>IActiveScript::InterruptThread</code> method and terminates the process. There is a callback hook. If the time-out is invoked, the <code>OnTimeOut</code> function is called to permit cleanup. Although it is possible to create infinite loops using this feature, it is more useful than harmful. |
| <code>//logo</code> | Displays an execution banner at execution time that is visible at the beginning of the <code>log.txt</code> log file. This is the default setting. |
| <code>//nologo</code> | Prevents display of the execution banner at execution time. |
| <code>//D</code> | Enables active debugging. |
| <code>//E:engine</code> | Use the engine to execute a script. |
| <code>//Job:xxxx</code> | Execute a WSF job. |
| <code>//X</code> | Execute the script in debugger. |

Using Parameters in WSH Tests

Parameters that are defined for a WSH test automatically add a name and value pair to the command line as an additional argument and set the parameters as environment variables for the called process. This functionality allows you to access all parameters defined for your test within the WSH script.

For example a WSH test is defined with `myscript.js` as script and `//B` as switch. Additionally the test requires a parameter called `IPAddress` with the value `192.168.1.5` and another parameter called `Port` with the value `1492`. The resulting command line for the WSH execution in this example is:

```
csript myscript.js //B IPAddress=192.168.1.5 Port=1492
```

Viewing Information Returned from WSH

To collect results of a WSH execution, the WSH script must generate a file called `output.xml` in the current working directory of the WSH test. All files residing in this directory are stored in the database and are downloadable through the list of files for the test execution. Files are excluded from storage when their extensions are defined under the file extensions to ignore in results property in the **Projects** area.



Note: The current working directory is dynamically created for each WSH execution. Do not use an absolute path when creating the file. Any relative path used will correctly refer to the current working directory.

Any information that a script writes to the WSH standard output goes into the `log.txt` text file that resides in the current working directory. This file is stored in the database and can be viewed as it is included in the file list of the test execution.

The following example shows how to print log information from a script:

```
WScript.Echo "This info will be written to the log.txt file"
```

The XML structure of `output.xml` begins with an element `ResultElement` that defines an attribute named `TestItem`, which specifies the name of the `ResultElement`.

The `ResultElement` must contain an element named `ErrorCount`, optionally an element named `WarningCount`, and a list of Incident elements.

The `ErrorCount` and `WarningCount` elements must contain a positive number or zero. The `ErrorCount` and `WarningCount` of the top-level `ResultElement` are used for evaluating success conditions, which determine

if a test has passed or failed. The XML file might contain additional elements that are not visible in the Silk Central GUI. The `output.xml` file is however stored in the database and is viewable as it is included in the file list of the executed test.

The *Incident* element represents an event that happened during the execution of the WSH test. *Message* and *Severity* are shown in the messages list of test executions in the Silk Central GUI. An *Incident* element must contain a *Message* and a *Severity* element.

The *Severity* element must hold one of the following values:

- Info
- Warning
- Error (or Exception)
- Failure

You can store additional information in the result file. The *ResultElement* may contain any number of sub-*ResultElements*, so information can be easily grouped. Sub-*ResultElements* make the result file easier to read. For compatibility reasons related to unit tests, JUnit and NUnit, *ResultElement* can be named *TestSuite* or *Test*.

The *ResultElement* may contain the following additional elements:

- *FailureCount*, which is treated the same way as error count
- *RunCount*, if a test is run multiple times
- *Timer*, for example for the duration of the test
- *WasSuccess*, for compatibility with NUnit result files
- *Asserts*, for compatibility with NUnit result files

The *Incident* element may contain a list of *Detail* elements.

The *Detail* element represents detailed information about an *Incident*. It must define a *TestName* element and an *Info* element. The *TestName* is used to give detailed information about where the *Incident* happened. The *Info* element holds detailed information about the *Incident*, for example a stack trace.



Note: Up through Silk Central 8.1, the value of the *Message* and *Info* elements had to be URL encoded (ISO-8859-1). Since version 8.1.1, URL encoding is no longer allowed.

Sample Result File

```
<ResultElement TestItem="WshOutputTest">
  <ErrorCount>1</ErrorCount>
  <WarningCount>1</WarningCount>
  <Incident>
    <Message>some unexpected result</Message>
    <Severity>Error</Severity>
    <Detail>
      <TestName>function main()</TestName>
      <Info>some additional info; eg. stacktrace</Info>
    </Detail>
  </Incident>
  <Incident>
    <Message>some warning message</Message>
    <Severity>Warning</Severity>
    <Detail>
      <TestName>function main()</TestName>
      <Info>some additional info; eg. stacktrace</Info>
    </Detail>
  </Incident>
</ResultElement>
```

Java Script Sample

The following script was used to generate the sample result file. To try this script save it with the extension .js.

```
function dumpOutput(dumpFile)
{
    dumpFile.WriteLine("<ResultElement TestItem=\"WshOutputTest
\">");
    dumpFile.WriteLine("    <ErrorCount>1</ErrorCount>");
    dumpFile.WriteLine("    <WarningCount>1</WarningCount>");
    dumpFile.WriteLine("    <Incident>");
    dumpFile.WriteLine("        <Message>some unexpected result</
Message>");
    dumpFile.WriteLine("        <Severity>Error</Severity>");
    dumpFile.WriteLine("        <Detail>");
    dumpFile.WriteLine("            <TestName>function main(</
TestName>");
    dumpFile.WriteLine("                <Info>some additional info; eg.
stacktrace</Info>");
    dumpFile.WriteLine("            </Detail>");
    dumpFile.WriteLine("        </Incident>");
    dumpFile.WriteLine("    <Incident>");
    dumpFile.WriteLine("        <Message>some warning message</
Message>");
    dumpFile.WriteLine("        <Severity>Warning</Severity>");
    dumpFile.WriteLine("        <Detail>");
    dumpFile.WriteLine("            <TestName>function main(</
TestName>");
    dumpFile.WriteLine("                <Info>some additional info; eg.
stacktrace</Info>");
    dumpFile.WriteLine("            </Detail>");
    dumpFile.WriteLine("        </Incident>");
    dumpFile.WriteLine("</ResultElement>");
}

function main()
{
    var outFile;
    var fso;
    fso = WScript.CreateObject("Scripting.FileSystemObject");
    outFile = fso.CreateTextFile("output.xml", true, true);
    outFile.WriteLine("<?xml version=\"1.0\" encoding=\"UTF-16\"?
>");

    dumpOutput(outFile);
    outFile.Close();
    WScript.Echo("Test is completed");
}

main();
WScript.Quit(0);
```

Visual Basic Script Sample

The following Visual Basic script also generates the sample result file, and saves it as Output.xml. To try this script save it with the extension .vbs.

```
WScript.Echo "starting"

Dim outFile
```

```

Dim errCnt
Dim warningCnt

outFile = "output.xml"
errCnt = 1 ' retrieve that from your test results
warningCnt = 1 ' retrieve that from your test results

Set FSO = CreateObject("Scripting.FileSystemObject")
Set oTX = FSO.OpenTextFile(outFile, 2, True, -1) ' args: file,
8=append/2=overwrite, create, ASCII

oTX.WriteLine("<?xml version=""1.0"" encoding=""UTF-16""?>")
oTX.WriteLine("<ResultElement TestItem=""PerlTest"">")
oTX.WriteLine("  <ErrorCount>" & errCnt & "</ErrorCount>")
oTX.WriteLine("  <WarningCount>" & warningCnt & "</
WarningCount>")
oTX.WriteLine("    <Incident>")
oTX.WriteLine("      <Message>some unexpected result</Message>")
oTX.WriteLine("      <Severity>Error</Severity>")
oTX.WriteLine("      <Detail>")
oTX.WriteLine("        <TestName>function main()</TestName>")
oTX.WriteLine("        <Info>some additional info; eg.
stacktrace</Info>")
oTX.WriteLine("      </Detail>")
oTX.WriteLine("    </Incident>")
oTX.WriteLine("    <Incident>")
oTX.WriteLine("      <Message>some warning message</Message>")
oTX.WriteLine("      <Severity>Warning</Severity>")
oTX.WriteLine("      <Detail>")
oTX.WriteLine("        <TestName>function main()</TestName>")
oTX.WriteLine("        <Info>some additional info; eg.
stacktrace</Info>")
oTX.WriteLine("      </Detail>")
oTX.WriteLine("    </Incident>")
oTX.WriteLine("</ResultElement>")

```

Shared Tests

Recurring tests and their requirement assignments can be managed centrally and can be shared and consumed across projects.

If your project is configured to consume tests from a specific project that is sharing tests (see [Adding Projects](#)), you can reuse these tests into your project by selecting them through the selection UI.

Sharing and Consuming Tests

Manage recurring tests and their requirement assignments centrally and share and consume these tests across projects.

If your project is configured to consume tests from a specific project that is sharing tests (see [Adding Projects](#)), you can reuse these tests into your project by selecting them through the selection UI.

To insert shared tests into your project:

1. In the menu, click **Tests > Details View**.
2. Right-click on the test container or the test folder into which you want to insert the shared tests.
3. Select **Shared Tests**.
4. Select **Insert Shared Tests**. The **Browse Tests** dialog appears.
5. Select the tests that you want to insert.



Tip: To narrow down the displayed test set, either set some criteria in the **Search** field or select a pre-defined filter from the list.

6. Click **OK**. The selected tests are inserted. The test properties contain a **Shared Test Reference**, which means a link to the shared test.
7. To detach a consuming test from a shared test, click **X** in the **Shared Test Reference** field.



Note: To remove a reference to a shared test, you require the **Tests and Executions - Manage assets** permission.

After removing the reference to a shared test, the test will no longer have any connection to the shared test. Updates will therefore no longer be pushed and received between the shared test and the consuming test.

To see all tests that are consume a specific shared test, click **Tests > Details View** in the menu and select the **Usages** tab.

When consuming tests, consider the following:

- The project that consumes the tests must be configured to reuse tests from the project that is sharing the tests.
- You can only insert tests into an existing test container.
- The actions **Accept Changes** and **Push Changes to Shared Test** handle differing test types between a consuming test and the corresponding shared test, by resulting in the same test type being used in both tests.
- When accepting changes from a test which uses shared steps, keywords, business components, or flows, the corresponding libraries must be visible in both the project that consumes the test and the project that shares the test.
- To retain assigned requirements, the project that consumes a test must either use the same requirements integration or the same Word or Excel import file as the project that is sharing the test.
- You can only update a requirement from Excel for which a unique external ID is defined in the Excel import file. Silk Central requires this ID to identify which requirement needs to be updated.
- Data-driven children, test package children, and linked test containers cannot be shared.
- Inherited test properties, parameters, and attributes are not considered in consuming tests. Such tests will inherit what is inherited in the consuming project. Accepting changes from the original test can lead to properties, parameters, and attributes being removed in consuming tests if inheritance is being used, as these properties, parameters, and attributes might not be available in the consuming project.
- Assigned issues are not copied to consuming tests.
- If you use test attributes, they must have the same name and type in both projects.
- When copying or baselining a consuming project, the references to the project that is sharing tests are retained. However, these references are deleted when exporting a consuming project.
- The bulk export/update interface and the web services do not yet consider links to original assets.
- You can also insert folders, including their complete hierarchy.

Maintaining Versions of Shared Tests - Accept and Push Changes

Maintaining recurring tests and their requirement assignments in one project and sharing them across multiple other projects saves time compared to maintaining such tests in each project individually. Maintaining tests in this case means keeping your tests up to date by adjusting descriptions, adding new steps, updating scripts, changing requirement assignments, and so on. To distribute such changes to your consuming tests you have to create a new test version for your shared tests. For additional information, see [Creating a Version](#). When creating a new version, all consuming tests in all consuming projects are flagged, which means the tests are marked with a blue arrow pointing up, to indicate that updates are available for these tests. Right-click such marked consuming tests and select **Shared Tests > Accept Changes** to update the tests to the latest version, or select **Shared Tests > Skip Changes** to keep the current version and to remove the update flag. Consuming tests will get flagged again with each new version for a shared test.

You may also face the scenario where you update the consuming test, and you then want to push those changes back to the shared test. To do so, click **Push Changes to Shared Test** in the **History** tab of your consuming test. Pushing the changes creates the following:

- A new version for the consuming test.
- A restore point for the shared test, which means a new version for the current state of the shared test.
- A new version for the shared test including the pushed changes.

Other consuming tests will then receive an update notification for the changes in the shared test.

Sharing Tests Out of a Consuming Project

Share tests out of a consuming project.

When you have extended the test set in a project that consumes tests from a project sharing tests, you can make these new tests available to other consuming projects by adding the tests to the shared test set in the sharing project.

1. In the menu, click **Projects > Project List**.
2. Select the consuming project that contains the new tests.
3. In the menu, click **Tests > Details View**.
4. Right-click on the test folder or the test that you want to share.
5. Select **Shared Tests > Share Tests**.



Note: This action is only available if the selection does not contain any tests that are already shared.

6. Select the target folder within the sharing project, where you want to insert the selected folders or tests.
7. Click **OK**. The selected test folders or tests are copied as a child to the selected folder in the sharing project.

Editing Tests

To edit a test:

1. In the menu, click **Tests > Details View**.
2. Select the test or the test package that you want to edit.
3. Click  (**Edit**) on the toolbar.

You can also click the **Properties** tab and click **Edit** or right-click the test or test package and select **Edit**.

The **Edit Test** dialog box appears.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

4. Edit the name and description of the selected test.
If the selected test is a test package, the **Update Package Structure on Result** check box is available. Check the **Update Package Structure on Result** check box if you want to update the structure of the test package according to the results of the test execution run.
5. Configure the properties of the test or the test package according to the test type as described in *Configuring Test Properties*.

Test Export, Update, and Import

The Silk Central integration with Microsoft Excel provides two options to modify your tests in Excel.

- *Export and update:* This option allows you to export manual tests from Silk Central to Excel, to update the tests in Excel, and to re-import the changed tests into Silk Central.

- *Import*: This options allows you to create manual tests and business-process tests in Excel and to import these tests into Silk Central.

In both scenarios, you have to create a mapping file to synchronize the columns between Excel and Silk Central.

 **Note:** When importing or updating tests from or when exporting tests to Excel, Silk Central supports only `.xlsx` files.

Rules for Exporting and Importing Tests

The following rules apply when importing or updating tests from Excel, or when exporting tests to Excel.

General Rules

- If a test row in Excel does not have a `TestID` or a **Path**, then it is considered to be a new test. The new test is created in a new folder in Silk Central, located under the test container from which the import functionality was executed.
- In Excel, you can the specify the **TestID** of an existing test in Silk Central to update the corresponding test in Silk Central when importing the Excel file.
- You cannot import or update automated tests from Excel.
- Attribute values that are inherited by a test are not exported to Excel when exporting the test.
- If you move a test to a different folder in Excel, the move is ignored when updating the tests in Silk Central from Excel.
- If you remove a test in Excel, the test will not be deleted when updating the tests in Silk Central from Excel.
- Use the **Path** column in Excel to specify the folder names for the new tests in Silk Central. When importing the Excel file into Silk Central, the new tests will be created in the specified folders.
- You cannot use `A-Z` as a customized column name. These are reserved for Excel's `A-Z`, which can be used in mapping.

Shared Steps

- To add a new shared step in Silk Central when importing or updating from Excel, specify the **SharedStepID** in Excel.
- You can delete and reorder shared steps in Silk Central by reordering or deleting the corresponding rows in Excel.
- You cannot modify existing shared steps by importing or updating them from Excel.
- If a `SharedStepID` that is specified in Excel does not exist in Silk Central, an error is reported.

Data-Driven Tests

- You can export data-driven tests to Excel and you can import data-driven tests from Excel.
- You cannot export non-atomic data-driven child tests from Silk Central to Excel.
- The data-driven properties of data-driven tests cannot be exported to or imported from Excel.
- You cannot update folders in Silk Central, which contain data-driven tests, from Excel.

Exporting a Test to Excel

1. In the menu, click **Tests > Details View** .
2. Right click a test container or test folder that includes the tests that you want to export.
3. Select **Export to Excel...** The **Export Tests to Excel File** dialog box displays.
4. Select the mapping file from the **Choose Mapping File** list box.

The default mapping file is `TestPropertiesDefaultMapping.xml`. For information about creating customized mapping files, see *Editing the Test Mapping File*.

5. *Optional:* Click  to edit the mapping file. Click  to delete the mapping file.

 **Note:** These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.

6. Click **OK**.

Test information is saved into the Excel file for you to edit.

Updating a Test from Excel

Before you can update a test from Excel, you first need to export the test from Silk Central.

1. In the menu, click **Tests > Details View**.
2. Right click a test container or folder that includes the tests that you want to update.
3. Select **Update from Excel...** The **Update Tests from Excel File** dialog box opens.
4. Select the mapping file from the **Choose Mapping File** list box.

The default mapping file is `TestPropertiesDefaultMapping.xml`. For information about creating customized mapping files, see *Editing the Test Mapping File*.

5. *Optional:* Click  to edit the mapping file. Click  to delete the mapping file.

 **Note:** These actions require the permissions **Modify Office import mapping files/Delete Office import mapping files**, respectively.

6. Select the Excel file in the **Test File (.xlsx)** field.
7. Click **OK**.

Test Import

This section describes how to import tests from Microsoft Excel into Silk Central.

Silk Central enables you to import tests from Microsoft Excel (Excel) into any container or folder in the **Tests** area and then maintain and execute all included tests in Silk Central. The import is a one-time event.

Before you can import tests, you need to create a test file in Excel. For more information, see *Test File*.

Silk Central uses the information stored in the Excel file and in a mapping file to determine what data to store. Silk Central uses a mapping file to map the tests in the Excel file to the tests in the **Tests** area. A default mapping file named `TestPropertiesDefaultMapping.xml` is preselected for the import. You can choose your own customized mapping file to map your tests. For more information about mapping files, see *Editing the Test Mapping File*.

Test File

You can create a test file in Excel to import manual tests and business-process tests into Silk Central. In the test file, you can define all tests and test steps, along with their properties, attributes, and parameters.

To map the tests in your test file to the tests in Silk Central, use an XML mapping file. The mapping file defines the location of the elements in your test file to Silk Central. You can give the items included in the test file any name you like, as long as you map them appropriately in your mapping file. For more information about mapping files, see *Editing the Test Mapping File*.

 **Note:** When importing or updating tests from or when exporting tests to Excel, Silk Central supports only `.xlsx` files.

Property Mappings

Click **Edit Mapping** on the **Import Tests from Excel File** or **Update Tests from Excel File** dialog to map the tests in your Excel file to Silk Central as follows:

- **Header Row:** Specify the row in your Excel file which contains the column names to be used for mapping the properties.
- **First Data Row:** Specify the row that contains the first test. This row and all rows below are interpreted as tests.
- **Separator Characters:** Specify the characters that should be interpreted as separators within property values. This is used for multi-list properties.
- **Name, Description:** Specify the column names in the header row that are mapped to test name and description, respectively.
- **Planned Time:** The expected amount of time for this manual test to execute.
- **Step** properties: The name, action description, and expected result of the test step.
- **Assigned Requirement IDs:** If requirements should be assigned to the test, specify the ID's of those requirements as comma-separated list. Make sure that this column is formatted as "Text" in Excel, otherwise the assignment may not work.
- **Business Component:** Only available for business process tests. If a business component is referenced by the test then the library is checked if a business component with this name exists in the library. If no such business component exists, a new business component with this name and steps is created in the library.
- **Assigned Testers:** Only available for business process tests. Specify the testers that should execute the test.
- **Behavior on Failure:** Only available for business process tests. Specify whether the test should stop or continue when a business component fails during execution.

Editing the Test Mapping File

Edit the test mapping file to import and update manual tests and business process tests from Excel.

The mapping file is used during the update process of export-update and during the import process.

You need the **Modify Office import mapping files** permission to edit the mapping file.

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, right-click the folder or container that you want to use as the parent node of the imported test and choose **Import from Excel File**. The **Import Tests from Excel File** dialog box opens.
3. Click **Edit Mapping**. The **Edit Mapping** dialog box opens.
4. Edit the variables of the test file in the fields.



Note: For information on the variables included in the test file, see *Test File*.

5. *Optional:* Edit the name of the mapping file.
6. Click **OK** to save your changes or click **Cancel** to cancel your changes.

Test Versioning

Silk Central creates new versions of tests during the import and update processes.

When you perform an update from Excel into Silk Central, two new versions of the test tree are created. The versioning process will record a snapshot before and after the update process.

Export-Update

Tests will contain default comments to note that they were updated from Excel if they are added or modified.

Import

Tests will contain default comments of `Created`.

Printing Tests

To print tests:

1. In the menu, click **Tests > Grid View** .
2. Use **Ctrl+Click** or **Shift+Click** to select the tests that you want to print.
3. Right-click the selection and choose **Print Test Details**. The *Test Details* dialog box appears.
4. Click **Print** or **Download as PDF**.



Tip: You can also select tests or folders in the **Tests** tree and select **Print Test Details**.

Test Runs Page

Tests > Details View > <Test> > Runs

The **Runs** page is available on test nodes in **Details View** and offers a listing of test execution results for the selected test.

The data grid representation of the **Runs** page facilitates the filtering, sorting, and grouping of large numbers of test runs.

For each run, the page displays the following columns:

| Column | Description |
|-----------------------|---|
| Actions | Actions that you can perform on the test run. |
| | New Issue Click to open the New Issue dialog box and create a new issue for the test. |
| | Assign Existing Issues Assign issues from an externally-configured issue-tracking system to the test. |
| Status | Status of the execution. For tests that are part of a running execution plan, the status is updated in response to the current status of the test run. If the current run is aborted, the status is reset to the status before the run. |
| Status Reason | The reason why a specific test run has the status Passed , Failed or Not Executed . |
| Run ID | The ID of the test run. Click to open the Test Run Results dialog box. If the test is running, click to view details of the execution. |
| Run Type | The Run Type column shows the test type during each run. The test type might change between two runs, for example when you convert the test from manual to automated. |
| Start Time | Time the run started. If the test is a manual test and currently running, Silk Central adds (<i>Running</i>) to the date and time. |
| Duration | Duration of the test run in <i>h/mm/ss</i> . |
| Execution Plan | The name of the assigned execution plan, or <i>Unassigned Tests</i> if the execution was a trial run or results were uploaded. Click to open the execution plan. |
| Executed By | The execution server from which the test was run. |
| Issues Found | Displays the amount of issues that are assigned to the test run. When no issues are assigned to the test run, the column is empty. Click on the link to access the issue in the Issues page of Tests > Details View . |
| Errors | Number of errors that were generated during the run. |

| Column | Description |
|------------------------------|---|
| Warnings | Number of warnings that were generated during the run. |
| Version | Version that the test was run against. |
| Build | Build number that the test was run against. |
| Execution Plan Parent | The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed. |
| Run Comment | For Silk Performer test runs, Silk Performer uses this column to add information to the test run when uploading results. For all other test types, you can use this column to add information to the run. |

To compare two or more test runs, use **Ctrl** or **Shift** to select runs. Right click on your selection and click **Compare Automated Test Results** or **Reports > Test Run Comparison** .

Test Issues Page

Tests > Details View > <Test Element> > Issues

The **Issues** page enables you to enter and track issues related to the selected test, container, or folder.

- New Issue** Click to assign new issues to the selected test. This button is only displayed if the currently selected object is a test.
- Assign Existing Issues** Click to assign issues from an external issue tracking system to the selected test. This button is only displayed if the currently selected object is a test.
- Update Issue States** Click to update issues states.

For each issue, the page displays the following columns:

| Column | Description |
|----------------|---|
| Actions | Actions that can be performed on the issue. Click  to delete the issue. External issues will not be deleted. |
| Issue ID | ID that has been automatically assigned to the issue. Click to edit the issue. |
| Assigned Test | Test that has been assigned to the issue. This column is only displayed if the currently selected object is a container or a folder. |
| Synopsis | Synopsis of the issue. |
| Status | Status of the issue. |
| External ID | Indicates if the issue is tracked by an external issue tracking system. Click an external issue number to link directly to the external issue tracking system. |
| Priority | The priority of the issue as defined in the external issue tracking system. The following issue tracking systems support this feature: Atlassian Jira Software, Bugzilla, Silk Central Issue Manager, and StarTeam. |

| Column | Description |
|------------|--|
| Test Run | The ID of the test run that the issue is assigned to. Click on the ID to access the Details page of the Test Run Results dialog box in the Executions area. |
| Created On | When the issue was created. |
| Created By | User who created the issue. |
| Profile | Name of the issue tracking profile, as defined in Silk Central > Settings > Issue Tracking . |

Versions

A version is a revision of an element in a certain state.

The following elements have versions:

- Tests.



Note: Data sets and inherited test properties, parameters, and attributes are not included in test versions.

- Shared objects.

You can manually create versions or revert to previous versions. All versions of an element are listed on the **History** page of the element. When you copy a project, the versions are also copied. If you are working with shared tests, see also [Maintaining Versions of Shared Tests](#).

Creating a Version

Create a new version of an element to track or to revert any changes to the element.

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click  in the toolbar.
This action is only enabled if the shared object or the test was changed.
The **Create Version** dialog box appears.
4. *Optional:* Enter a comment on the version in the **Comment** field.
5. Click **Ok**. The new version is shown in the **History** page of the element.

Viewing a Version

To see if a specific version of a shared object or a test applies to your testing needs, view the details of the version.

To view the details of a version of a shared object or a test:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click the **History** tab.
4. Right click on the version of which you want to see the details.
5. Click **View version <ID>**.
6. The **View Version** dialog box opens. If the version includes attachments, you can view the attachments by clicking on their number in the **Attachments** column of the **Steps** section.



Note: Use the arrows to change the version number and view additional versions.

Comparing Versions

To view the differences between two versions of a shared object or a test, compare the versions.

To compare two versions of a shared object or a test:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click the **History** tab.
4. Select the two versions you want to compare with **Ctrl+Click**.
5. Right click on the selection.
6. Select **Compare Version <x> and <y>**. The **Compare Versions** dialog box opens, displaying all the added, changed, and removed content.



Note: Use the arrows to change the version numbers and compare additional versions.

Reverting to a Previous Version

To undo changes to a shared object or a test, revert to a previous version. Note the following special cases when reverting to a previous version:

- If the value of an inherited attribute has changed compared to the current version, the most current value is again inherited in the reverted version.
- If the type of an attribute has changed compared to the current version, the attribute is ignored in the reverted version.
- If the value of a list-type attribute no longer exists in the current version, the original value is added to the attribute again and used in the reverted version.

To revert a shared object or a test to a previous version:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click the **History** tab.
4. Right-click on the version to which you want to revert the shared object or test to.
5. Click **Revert to Version <ID>**.
6. The shared object or test is reverted to the selected version and a new version is created in the **History** page. Attachments and data sets are not reverted for tests.

Viewing Recent Changes

To view recent changes to tests:

1. In the menu, click **Tests > Details View**.
2. Click  to filter out all test elements except those that have been changed since your last change acknowledgement.



Note: The recent changes filter is selected automatically in the **Filter** list box.

3. When you have reviewed the changes, you can accept them by clicking .
The acknowledge function resets the recent changes filter.

4. Click  again to remove filtering and see all tests.



Note: All test changes generate time-stamped entries in the test history.

Tracking the History of a Test

Silk Central provides a complete history of all changes that are made to tests. History information is read-only, and cannot be edited or permanently deleted.

To track a test element's history:

1. In the menu, click **Tests > Details View**.
2. Select the test in the **Tests** tree.
3. Click the **History** tab.

The properties of the test are displayed in tabular format.

Test History Page

Tests > Details View > <Test Element> > History

The **History** page details the revision history of the selected test, test container, folder, or project. It also includes an entry for every project baseline that includes the selected element, with links to the baseline and the corresponding element in the baseline. If the selected element was created as part of a project baseline, the first entry in the **History** page includes links to the original project and the corresponding element in the original project. You can use the links only to access active projects or baselines. For more information on project baselines, see the *Administration* topics in this Help.

The **History** page additionally includes an entry for each version of the selected test, and allows you to create new versions, revert to a previous version, and view the changes between two selected versions.

For each change, the page displays the following columns:

| Column | Description |
|-------------------|--|
| Actions | Only for version entries. The actions you can perform are View Version , Revert to Version and Delete Version . |
| Version | Only for version entries. The version of the test. |
| Comment | A comment on the changes. |
| Created On | Date and time of the change. |
| Created By | User that made the change. |



Note: When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

Configuring Test Properties

Configure test properties like attributes, parameters, attachments, and success conditions.

Test Properties

Tests > Details View > Properties

The **Properties** page offers detail on all properties and relevant information for the selected test, test folder, test package, or test container. You have the option of expanding and collapsing the information in

each section on this page. This enables you to customize the information you see to your specific needs. For test nodes, the following properties are configured when tests are created:

| Property | Description | | | | | | | | | | |
|------------------------------|--|--------|-------------|------------------|---|---------------|--|-----------------------|--|------------------------------|---|
| Test Name | Name of the test. | | | | | | | | | | |
| Test ID | Database identifier of the test. | | | | | | | | | | |
| Description | <p>Description of the test.</p> <p> Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for Description fields.</p> | | | | | | | | | | |
| Status | <p>For tests or test packages that are included in a configuration suite, the status is an aggregation of all statuses of all configurations within the suite:</p> <ul style="list-style-type: none"> • If the test is assigned only to the configuration suite, the status is <code>Passed</code> only if it is passed in all configurations. If the status is not <code>Passed</code> in all configurations, the status of the worst run is propagated, with the following top-down priority: <ol style="list-style-type: none"> 1. N/A 2. Not Executed 3. Failed 4. Passed <p>The status of a requirement that is assigned to the test is <code>Failed</code> if the test fails in one or more of the configurations.</p> • We do not recommend assigning the test to a configuration suite and one or more simple execution plans. However, in this case, the aggregate of the configuration statuses is one status, and the last statuses of the execution plans are other statuses. The status that was executed last determines the last status of the test. • If the test is assigned to a simple execution plan only, the status is updated in response to the status of the last test run. | | | | | | | | | | |
| Last Execution | Last time this test was executed. For tests that are part of a running execution plan, the last execution is updated based on the current test run. | | | | | | | | | | |
| Assigned Executions | <p>Lists all execution plans that the selected test is assigned to. For each execution plan in the list, the grid displays the following columns:</p> <table border="1"> <thead> <tr> <th>Column</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Execution</td> <td>Name of the execution plan. Click to view or edit the execution plan.</td> </tr> <tr> <td>Status</td> <td>The status of the last run of the test in the execution plan. The status is <code>Not Executed</code>, if the test was not executed in the last run of the execution plan. For data-driven tests or test packages, the status of all last runs is displayed in a bar graph.</td> </tr> <tr> <td>Last Execution</td> <td>Last time the test was executed as part of the execution plan.</td> </tr> <tr> <td>Execution Plan Parent</td> <td>The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed.</td> </tr> </tbody> </table> | Column | Description | Execution | Name of the execution plan. Click to view or edit the execution plan. | Status | The status of the last run of the test in the execution plan. The status is <code>Not Executed</code> , if the test was not executed in the last run of the execution plan. For data-driven tests or test packages, the status of all last runs is displayed in a bar graph. | Last Execution | Last time the test was executed as part of the execution plan. | Execution Plan Parent | The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed. |
| Column | Description | | | | | | | | | | |
| Execution | Name of the execution plan. Click to view or edit the execution plan. | | | | | | | | | | |
| Status | The status of the last run of the test in the execution plan. The status is <code>Not Executed</code> , if the test was not executed in the last run of the execution plan. For data-driven tests or test packages, the status of all last runs is displayed in a bar graph. | | | | | | | | | | |
| Last Execution | Last time the test was executed as part of the execution plan. | | | | | | | | | | |
| Execution Plan Parent | The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed. | | | | | | | | | | |
| Created On | Date and time the test was created. | | | | | | | | | | |
| Created By | Name of the user who created the test. | | | | | | | | | | |
| Changed On | Date and time the test was last changed. | | | | | | | | | | |

| Property | Description |
|-----------------------------------|---|
| Changed By | Name of the user who last changed the test. |
| Planned Time [hh:mm] | Planned execution time of the test. Only for manual tests. |
| Test Properties | Test properties that are specific to the test type. |
| Success Conditions | All success conditions that have been configured for the test. For test package nodes, all success conditions except the execution time-out are disabled and hidden. |
| Integration Default Folder | Shows the name of the default container or folder, where tests from external planning tools are created. For detailed information, see Setting a Test Node as Integration Default for External Agile Planning Tools . |

Test Attributes

Attributes are freely definable characteristics that can be applied to tests. You can filter the **Tests** area for the tests with a specific attribute and assign all such tests to execution plans. Examples include a platform attribute that can be applied to product components and a priority attribute that can be applied to tests.



Note: Inheritance of attributes is similar to inheritance of properties and success conditions. Attributes that are assigned to a parent node are inherited throughout all sub-folders and child tests.

Assigning Attributes to Tests

To assign an attribute to a test:

1. In the menu, click **Tests > Details View** .
2. Select the test to which you want to assign an attribute.
3. In the **Attributes** section, click **Add Attribute**. The **Add Attributes** dialog box appears.
4. Click **+** in the **Add** column of the attribute that you want to assign. Based on the attribute type you have selected, an **Edit Attribute** dialog box appears, where you can specify which of the available attribute values you want to assign to the test.
5. Select an attribute value and click **OK** to assign the attribute to the test.

Editing Test Attributes

To edit a test attribute:

1. In the menu, click **Tests > Details View** .
2. Select the test for which you want to edit an assigned attribute.
3. In the **Attributes** section, click the **Edit** icon of the attribute you want to edit. The **Edit Attribute** dialog box displays. The available options in the **Edit Attribute** dialog box vary depending on the attribute type that you have selected.
4. Select the required value and click **OK** to save your settings.

Deleting Attributes from Tests

To delete an attribute from a test:

1. In the menu, click **Tests > Details View** .
2. Select the test from which you want to delete an assigned attribute.
3. In the **Attributes** section, click the **Delete** icon of the attribute you want to delete. The **Delete Attribute** confirmation dialog box displays.

4. Click **Yes** to delete the attribute.



Note: Inherited attributes cannot be deleted.

Test Parameters

Parameters are freely configurable input values that can be assigned to different test types and used in a variety of ways. They help to define tests by defining test data.

Test parameters that are contained within a property of a test, for example test-data for Silk Test Classic tests, are listed at the top. Unused parameters are appended to the bottom of the list and grayed out, analogous to a disabled state.

| | |
|------------------------------|---|
| Silk Performer | These tests use predefined parameters that represent the project attributes that are defined in a selected Silk Performer test. |
| JUnit, Silk4J | Any JUnit and Silk4J test class can access a custom parameter of the underlying test as a Java system property; the launcher passes these parameters to the executing virtual machine using the <code>-D</code> VM argument. Within the test the values of the parameter can be accessed using <code>System.getProperty("myParam")</code> . |
| NUnit, Silk4NET | For NUnit and Silk4NET each parameter will be set as an environment variable. Within the test the value of the parameter can be accessed using <code>Environment.GetEnvironmentVariable("myParam")</code> . |
| Silk Test Classic | Parameters can be passed to Silk Test Classic by specifying them in the Test Data field in the Test Properties of the Silk Test Classic test in Silk Central. For example, if you want to pass the values of your Silk Central parameters <code>Name</code> and <code>Password</code> to Silk Test Classic, specify <code>"\${Name}"</code> , <code>"\${Password}"</code> in the Test Data field in the Test Properties . Your Silk Test Classic script also requires two parameters, which will receive the specified values. Example: <pre>[-] testcase ProcessParams(STRING name, STRING password) [] Print(name) [] Print(password)</pre> |
| External Processes | For any test type where the execution takes place in an external process (WSH, Process Executor, NUnit, JUnit, MSTest, DotNetExplorer), each parameter will be set as an environment variable for the process. This is also the case if the parameter name matches the name of an operating system variable, so that the value of the system variable will be replaced by the parameter value, except when the parameter value is an empty string. |
| Silk Test Workbench | The parameter needs to be defined with the same name in the Silk Test Workbench test, to be set when being executed in Silk Central. |
| Configuration testing | You can override the test parameters in an execution plan. For additional information, see <i>Execution Plan Parameters</i> . |
| Custom test types | For details on custom test types, refer to the <i>Silk Central API Help</i> . |

Predefined Parameters

Silk Central provides information on execution plan run properties during test executions as parameters.

These parameters can be read by your test automation tool as environment variables (see [Test Parameters](#)). Additionally, any JUnit test class can access a test parameter of the underlying test as a Java system property (see [Accessing Silk Central Parameters through Java System Properties](#)). Use the `AttributeGet` methods to read these parameters in a Silk Performer script.

These parameter values can also be referenced in placeholders, as described in [Parameter-Token Replacement Upon Execution](#).

The following predefined parameters can be accessed:

| Parameter | Description |
|---|--|
| #sctm_execdef_name | The name of the execution plan that is currently executed. If the execution was triggered from the Tests area, the parameter has the value <code>Unassigned Tests</code> . |
| #sctm_execdef_id | The numeric identifier (database key) of the execution plan that is currently executed. |
| #sctm_product | The name of the product as defined in the executed test container. |
| #sctm_version | The name of the version to which the results of the execution are associated. |
| #sctm_build | The name of the version to which the results of the execution are associated. |
| #sctm_keywords | A comma separated list which contains the keywords that are defined for this execution plan. |
| #sctm_regular_execdef_run_id | The numeric identifier (database key) of the regular execution plan run. For setup and cleanup tests this property contains the run ID of the regular run. |
| #sctm_test_results_dir | The path of the directory where the result files of the test reside. |
| #sctm_test_name | The name of the Silk Central test being executed. |
| #sctm_test_id | The numeric identifier (database key) of the test being executed. |
| #sctm_source_root_dir | The local path of the test container's root directory, where all files under version-control reside. |
| #external_id | The unique identifier of a test case in a test package to be executed. You can use this as a filter if only specific test cases in a test package should be executed. |
| #is_test_package_root | If <code>true</code> , the test being executed is the root node of the test package (default). If <code>false</code> , only specific test cases in a test package specified by <code>#external_id</code> are executed. |
| #sctm_data_driven_parent_test_name | The name of the parent test, if the executed test is a data driven row. |
| #sctm_data_driven_parent_test_id | The numeric identifier (database key) of the parent test, if the executed test is a data driven row. |

Creating Custom Parameters

To create a custom parameter:

1. In the menu, click **Tests > Details View**.
2. Select the test node for which you want to create a new parameter.
3. In the **Parameters** section, click **Add Custom Parameter**. The **Set Custom Parameter** dialog box displays.

4. Provide a name for the parameter.
5. In the **Type** list box, select the parameter type.

- String
- Number
- Float
- Boolean
- Password
- Character

6. Define the parameter value that you want to assign to the selected test.



Note: Values for parameters of type String must be set in quotation marks (“”) if you want to use the parameter in Silk Test Classic executions.

7. Click **OK**.



Note: Parameters are automatically assigned to all sub-folders and child tests of the nodes to which they've been assigned.

Adding Predefined Parameters to Silk Performer Tests

To add a predefined parameter to a test:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the test node to which you want to add a predefined parameter.
3. In the **Parameters** section, click **Add Predefined Parameter** .



Note: The **Add Predefined Parameter** button is only available for Silk Performer tests, when the **Project** property is already defined.

The **Add Predefined Parameter** dialog box appears, which lists all of the project attributes that are available in the project file.

4. To add any of the listed parameters, click the corresponding add icon.
5. On the dialog box that appears, specify the actual value for the parameter.
6. Click **Save** to add the parameter to the active **Tests** tree node.

Editing Predefined Parameters

To edit a predefined parameter:

1. In the menu, click **Tests > Details View** .
2. Select the test node for which you are editing an existing parameter.
3. In the **Parameters** section, click the **Edit** icon of the parameter you want to edit. The **Set Custom Parameter** dialog box displays.
4. Edit the parameter values as required.



Note: Inherited parameters cannot be edited. Uncheck the **Inherit from parent** check box to enable editing of the parameter's **Value** setting. Parameter **Name** and **Type** settings cannot be edited.

Deleting Predefined Parameter Assignments

To delete a predefined parameter assignment:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the test node for which you are deleting the assignment of an existing parameter.

3. In the **Parameters** section, click **Delete** in the **Actions** column of the parameter that you want to delete.



Note: Inherited parameters cannot be deleted. Uncheck the **Inherit from parent** check box on the **Set Parameter** dialog box to enable deletion of an inherited parameter.

4. Click **Yes** on the **Delete Parameter** dialog box to delete the parameter.

Parameters within Parameter Values

You can use parameter values that contain parameters. The evaluation result of such parameter values is shown in a bold font in the GUI. The following example shows how to use such a parameter:

```
parameterA := aaa
parameterB := bbb + ${parameterA}
```

Evaluated values:

```
parameterA = aaa
parameterB = bbb + aaa
```

Parameter Notations

The following parameter notations are supported:

For all tests:

```
${<parameter>}
```

All characters are allowed for parameter names, except \$, {, }, and #.

Deprecated notation for manual tests:

```
#<parameter>#
```

For manual tests, the following characters are allowed for parameter names: 0-9, a-z, A-Z, and _.

Additional notation for Silk Test Classic tests:

```
${<parameter>}
```

For Silk Test Classic tests, the following characters are allowed for parameter names: 0-9, a-z, A-Z, and _.

Parameter-Token Replacement Upon Execution

Any string input for a property of a test may contain placeholders in the following form: \$

{parametername}. parametername must match the name of a parameter defined or inherited for the test. At execution time, the placeholder is replaced by the value entered for the parameter with the denoted name. This makes recurring strings in properties more customizable and facilitates the editing of common definitions.

When Silk Central finds a parameter with the notation \${parametername}, it first checks if the parameter is included in the defined parameters, and if not, it checks if the parameter is defined as an environmental variable on the execution server. Environment variable values are used as defined at the startup time of the execution server. The Silk Central variable name only matches the name of the environment variable if the case is exactly the same, or if all characters are either lowercase or uppercase. Example: Silk Central matches the environment variable SilkHome against \${SilkHome}, \${silkhme} and \${SILKHOME}, but not against \${silkhme}.

For example, if the value of a JUnit classpath property is junit.jar;\${MyWorkingDir}/myclasses and the parameter MyWorkingDir has the value C:/Temp/MyWorking, the resultant effective property value is junit.jar;C:/Temp/MyWorking/myclasses.



Note: The value of a parameter may also contain other parameter placeholders, which allows nesting based on the same principle.

Test Parameters Page - Silk Test Workbench

Tests > Details View > <Test Element> > Data Set

When passing parameters from a Silk Central test to a Silk Test Workbench test, these parameters must be defined as input parameters in the Silk Test Workbench script and they must have the same name in Silk Central as in the script. The following data types should be used:

| Silk Central | Silk Test Workbench |
|--------------|----------------------|
| String | Text |
| Number | Number (Long) |
| Number | Number (Long Long) |
| Number | Enumeration |
| Float | Number (Double) |
| Boolean | Boolean (True/False) |

Predefined parameter names in Silk Central start with a hash while Silk Test Workbench does not allow parameter names starting with a hash. To pass a predefined parameter from Silk Central to a Silk Test Workbench test, you have to create an additional parameter without the hash in the name in Silk Central and map the predefined parameter to the new parameter.

For example, to pass the predefined parameter `#sctm_test_results_dir` to a Silk Test Workbench script, you have to apply the following workaround:

1. Specify an input parameter in the Silk Test Workbench script with the name `resultsdir`.
2. Specify a parameter in Silk Central with the name `resultsdir` and the value `${#sctm_test_results_dir}`.

Silk Central will then replace the placeholder with the actual directory and pass it to the Silk Test Workbench script.

Test Attachments

The **Attachments** section lists all files and links that have been uploaded as attachments to the selected test container, folder, or test.

Check the **Include Child Attachments** check box to additionally display all attachments of child tests, folders, and test containers of the selected node.

Attaching Files to Test Elements

To attach a file to a test element:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select a container, folder, or test.
3. In the **Attachments** section, click **Upload File**. The **Upload File** dialog box appears.
4. Click **Choose File** to select a file from your local file system.
5. *Optional:* Enter a **Description** for the attachment.
6. Click **OK** to upload the attachment to the server and associate it with the selected element.

Attaching Links to Test Elements

To attach a link to a test element:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a container, folder, or test.
3. In the **Attachments** section, click **Attach Link**. The **Attach Link** dialog box appears.
4. Type the URL in the **Link** field.
5. *Optional:* Type a **Description** for the attached link.
6. Click **OK** to associate the link with the selected element.

Viewing Test Attachments

For a selected test element, the **Attachments** page lists the attached files and links. The attachments are by default displayed in the order in which they are uploaded, but you can sort the attachments by columns. To display the attachments that are associated with child elements of the selected element, check the **Include Child Attachments** check box.

To view a test attachment:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the element for which you want to view an attachment.
3. In the **Attachments** section, click on the name of the attachment.

The name of each listed attachment serves as a link. File-attachment links open a **Save As** dialog box, enabling you to download the attachment to your local file system. Link-attachments link directly to the link destinations in a new browser window.

Downloading Files from Test Elements

To download all files from a test element:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a container, folder, or test.
3. Click the **Attachments** tab.
4. In the **Attachments** section, click **Download All Files**. All attached files are downloaded to your local disk as a `zip` archive, depending on how you configured your browser's download settings.



Note: To prevent from excessive downloads, **Download All Files** is disabled if **Include Child Attachments** is checked.

Deleting Attachments from Test Elements

To delete an attachment from a test element:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the element from which you want to delete an attachment.
3. In the **Attachments** section, click  in the **Actions** column of the attachment you want to delete.
4. Click **Yes** on the confirmation dialog box to delete the attachment from the project.



Note: Only one attachment at a time can be deleted.

Data-Driven Tests

Data-driven tests are tests that are derived from values in an existing data source, such as a spreadsheet or a database. Before you can work with data-driven tests, you need to configure a data source. For detailed information on configuring data sources, see [Data Sources for Data-Driven Tests](#).

Creating Data-Driven Tests

To create a data-driven test:

1. In the menu, click **Tests > Details View** .

2. Create a new test.

For more information, see *Creating Tests*.

3. Click the **Properties** tab of the newly created test.

4. In the **Data-driven Properties** section, click **Edit**. The **Data-driven Properties** dialog box appears.

5. Select a preconfigured data source from the **Data Source** list. For detailed information on configuring data sources, see [Data Sources for Data-Driven Tests](#).



Note: Your data source may contain up to 100 rows. If it contains more than 100 rows, you need to enter a query on the **Data-driven Properties** dialog box that returns 100 rows at most. This prevents generating more than 100 tests out of one data source.

6. Click **Next** to continue.

7. Select a data set from the **Data Set** list.

In the case of Excel data sources, this is a worksheet name. In the case of database data sources, this is a table name.

8. Check the **Each data row is a single test** check box to have each row in your data set considered to be a separate test (non-atomic test), or do not check this check box to create a single test for all data rows of your data set (atomic test). For detailed information, see [Atomic and Non-Atomic Data-Driven Tests](#).

9. *Optional:* Enter a SQL query in the **Filter query** field to filter your data set based on an SQL-syntax query.

Only simple `WHERE` clause queries are supported. For example, to filter data to only include data rows that contain `Boston` in the `City` column, enter the following **Filter query**:

```
City = 'Boston'
```

10. Check the **Enable data-driven properties** check box to enable data-driven functionality. The configured data source is only used for this test if this setting is enabled.

11. Click **Finish** to save your settings.



Note: Data-driven property settings are visible in the **Data-driven Properties** section of each test's **Properties** page.



Note: To use the data-driven test functionality of Silk Central with Silk Performer scripts, data sources with column names matching the corresponding Silk Performer project attributes must be used in conjunction with `AttributeGet` methods.

Adding a Data Source Value to a Manual Test Step

To add a data source value to a manual test step:

1. In the menu, click **Tests > Details View** .

2. Create a new data-driven test.

Select `Manual` as the test type and configure test steps.



Note: To view the values included in your data source, expand the **Data-driven Properties** section on the **Properties** page of the test.

3. Click the **Steps** tab.

4. Select the test step that should reference the data source value.

5. In the **Action description** field, enter a parameter that references the relevant column in your data source, using the syntax `${<column name>}`.

For example, if you want a test step to retrieve password parameters from a spreadsheet that has a column called *Password*, you would write the parameter as `${Password}`. When you execute the manual test step, the parameter is replaced by an actual value in the corresponding data-driven data source.

Downloading CSV Data From a Data Source

To download CSV data from a data source:

1. In the menu, click **Tests > Details View**.
2. Select a test that relies on the data source from which you want to download data.
3. Click the **Properties** tab.
4. In the **Data-driven Properties** section, click  in the **Actions** column of either the data source or the data set, depending on which entity contains the data you want to download.
5. Specify the location on your local system to where the data is to be downloaded.
6. Click **OK** to download the data in CSV format.

Editing Data-Driven Properties

To edit data-driven properties:

1. In the menu, click **Tests > Details View**.
2. Select the test that has the property you want to edit.
3. Click the **Properties** tab.
4. In the **Data-driven Properties** section, click  in the **Actions** column of the property you want to edit.
5. Edit the property as required.
6. Click **OK** to save your changes.

Removing Data-Driven Properties

Remove all data-driven properties when the data-driven properties are no longer relevant, and when you want to delete a data source.

To remove data-driven properties:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select the node from which you want to remove the data-driven properties.
3. Click the **Properties** tab.
4. In the **Data-driven Properties** section, click **Remove**. All data-driven properties are removed from the node.



Note: The data-driven properties are also removed from all sub-nodes that inherit the properties of the node.

Atomic and Non-Atomic Data-Driven Tests

When planning data-driven tests, you should first be aware of the following two different data-driven test types that are available in Silk Central:

- | | |
|-------------------------------------|---|
| Atomic data-driven tests | A single test result is generated for all data rows of your data source. This means that the test is only successful if the execution with every single data row is successful. If the execution with one data row fails, the whole test is marked as failed. |
| Non-atomic data-driven tests | Each data row of your data source is represented by a test of its own. This means that each data row produces a failed or passed test result. For example, if your data source is |

a spreadsheet with four rows, you will have the original test you created as a parent test in addition to four new child tests, one for each of the data rows.

For detailed information on creating data-driven tests, see [Creating Data-Driven Tests](#).

The parent test created in this process does not have parameters associated with it, since it only represents a structuring instance for its child tests and no longer functions as an actual test. All values found in the data source will be listed in the parent test's **Data-driven Properties** section.

When assigning a parent test to a requirement, links to requirements are only inherited when using atomic data-driven tests.

You can not assign the parent test of a non-atomic data-driven test to a setup or cleanup test execution, as such a parent node is treated as a folder. You can assign one of its child nodes though, and you can also assign an atomic data-driven test to a setup or cleanup test execution.

For detailed information on data source limits, see [Data Source Limits](#).

Worksheet Handling

If your data source is a Microsoft Excel worksheet, you should follow these guidelines to ensure a successful and maintainable data-driven test setup:

- Make your column names self-describing. This will make the data source setup in Silk Central easier to maintain.
- If you use multiple worksheets, use consistent column names across the worksheets. This will make it easier for you to apply filters for selecting columns for your data source setup.
- Use certain columns as “key” columns. Key columns will allow you to maintain your data source file, while Silk Central is still able to identify specific data rows due to the value in the key column, despite changes in row orders. Values within a key column should be unique.

Data Import Considerations

When importing data rows from an external data source, Silk Central does not account for row sorting functionality used in the external data source. Due to this, the order of data rows in Silk Central might differ from the row order in the external data source. Silk Central also ignores any format settings that have been applied in the external data source. For example, if you formatted date cells in an Excel worksheet to display the date in a certain way, Silk Central will ignore this setting and import any date values in the base format "YYYY.MM.DD HH:MM:SS.M".



Note: Your data source may contain up to 100 rows. If it contains more than 100 rows, you need to enter a query on the **Data-driven Properties** dialog box that returns 100 rows at most. This inhibits to generate more than 100 tests out of one data source.

Success Conditions

Use success conditions to determine whether a test is successful or not. You can assign one or more success conditions to each test node or suite node in the **Tests** area. If a success condition is not met during the execution of the test it is assigned to, the execution run is marked as `Failed`, except for the execution time-out success condition, for which the test execution is marked as `Not Executed`. For a test package, all success conditions except the execution time-out are disabled and hidden.

In **Details View**, when you select a node in the **Tests** tree, the **Success Conditions** table on the **Properties** page of the node displays all success conditions that are configured for the selected node. The table includes the name of each condition, whether or not the condition is active, the maximal value of the condition, and whether or not the condition is inherited.

A success condition is only evaluated when it is active. To activate and deactivate success conditions see [Editing Success Conditions](#). The available types of success conditions differ depending on the test type. The following success conditions are currently available in Silk Central:

| Success Condition | Description |
|--|---|
| Errors Allowed | Maximal number of errors allowed for the test. This success condition is active by default. |
| Warnings Allowed | Maximal number of warnings allowed for the test. |
| Execution Time-Out [s] | Maximal time-out allowed for the test in seconds. |
| Page Time: Avg. Page Time [s] | Maximal allowed average time to load a page. |
| Page Time: Max. Page Time [s] | Maximal allowed maximum time to load a page. |
| Transaction Response Time: Avg. Trans(Busy)ok [s] | Maximal allowed average response time for a transaction in the test. |
| Transaction Response Time: Max. Trans(Busy)ok [s] | Maximal allowed maximum response time for a transaction in the test. |

The inheritance of success conditions is similar to inheritance of properties. Success conditions that are assigned to a parent node are inherited throughout all sub-folders and child tests.

Editing Success Conditions

In **Details View**, when you select a node in the **Tests** tree, the **Properties** page of the node displays the success conditions that are associated with the node. The **Success Conditions** table includes the name of each condition, whether or not the condition is active, the maximal value of the condition, and whether or not the condition is inherited.



Note: For test package nodes, all success conditions except the execution time-out are disabled and hidden.

To edit the success conditions of a test:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select a test container, test folder, or test.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click  to the right of the **Success Conditions**. The **Edit Success Conditions** dialog box displays.
5. Uncheck the **Inherit from parent** check box of any success condition you are editing.
6. Edit values as required.
7. Specify if conditions should be active or inactive by checking or un-checking their **Active** check boxes.
8. Click **OK** to save your settings.

Executing a Trial Run of a Test

You can perform a trial run of a test to check if the test works as intended.

To perform a trial run of a test:

1. In the menu, click **Tests > Details View**.
2. Right-click the test that you want to try out in the **Tests** tree.
3. Select **Trial Run**.
4. The **Go To Activities** dialog box appears. Click **Yes** if you want to analyze the results on the **Activities** page, or click **No** if you want to remain on the current Web page.

For additional information, see *Activities*.



Note: Check the **Don't show this dialog again (during this login session)** check box if you don't want to be asked about switching to the **Activities** page again in the future. This setting will be discarded when you log out of Silk Central.

After successfully completing a trial run, you are ready to start *planning your executions*.

Libraries

Libraries are collections of shared objects, which are objects that can be shared between multiple tests in multiple projects.

Shared objects reduce the amount of objects that a test engineer needs to maintain. Libraries are displayed, organized, and maintained through a hierarchical tree structure, called the **Libraries** tree. The **Libraries** tree enables you to organize shared objects in any number of hierarchy levels. You can edit or delete any node in the tree. Right-click on a node in the tree to cut it, or to copy and paste it to another location. Use the **Visibility** page to define for which projects the selected library is visible. The root node of the **Libraries** tree is called **Libraries** and cannot be edited.

The **Usages** page of each shared object in the tree lists the objects that use the selected object, for example keyword-driven tests, business process tests, shared objects, keywords, or keyword sequences.

The following object types can be included in a library:

- Business components. These can be used for business process testing.
- Flows. These can be used for business process testing.
- Shared steps. These can be used in manual tests or in business components.
- Keywords and keyword sequences. These can be used for keyword-driven testing.



Note: Objects in the **Libraries** unit can be created, edited, and viewed by the Test Manager, Tester, and SuperUser roles. The Test Manager and SuperUser roles can additionally delete objects from the **Libraries** unit. Users with the roles Project Manager, Analyst, or Reporter can view objects in the **Libraries** unit.

Creating a Library

1. In the menu, click **Tests > Libraries**.
2. In the **Libraries** tree, select the root node, which is called **Libraries**.
3. In the toolbar, click  to create a new library.



Note: You can also right click on the root node and click **New Library**.

Managing Keyword Libraries and Library Properties

Tests > Libraries > <Library> > Properties

To view or update the properties of a node in the **Libraries** tree, select the node and click the **Properties** tab.

To set the library as read-only, check **Read-Only**. While a library is set to read-only, the following actions are not enabled:

- Adding shared objects to the library.
- Editing shared objects and folders in the library.
- Uploading a keyword library in the UI, from Silk Test, or by using the Web Service API.
- Extracting shared objects from a test into the library.
- Combining shared objects in a test that is using the library.

In the **Keyword Library** section, click **Upload** or **Delete** to upload or remove an external keyword library.

Making Libraries Accessible to Projects

Tests > Libraries > <Library> > Visibility

On the **Visibility** page, you can make the selected library visible to all projects or limit its visibility to a selection of specific projects. To change the visibility of a the selected library to a project, click **Edit Visibility**.

The **Edit Visibility** dialog box displays the projects that can be assigned to the library by the current user. The dialog box includes the following items:

| Item | Description |
|--------------------------|--|
| All Projects | Click this option button to assign all projects to the library. |
| Selected Projects | Click this option button to assign selected projects from the list to the library. |
| Select All | Click to assign all listed projects to the library. |
| Deselect All | Click to unassign all listed projects from the library. |

 **Note:** Projects that are currently using the library are grayed out and you cannot unassign them from the library.

When a baseline of a project is created, which uses shared objects, the libraries that include these objects are by default visible to the new project baseline. For more information on project baselines, see the *Administration* topics in this Help.

Managing Keywords in a Library

The **Keywords** page offers two different views in **Tests > Libraries** , depending on the selected object.

Library

If your selected object is a library, all keywords and keyword sequences within this library are listed.

Libraries that contain uploaded keyword libraries are displayed with the symbol . The following actions are possible:

| Task | Steps |
|---|---|
| Implementing a keyword (adding code to be able to actually run the keyword in a test) | <p>Prerequisites: If your keyword contains parameters, you need to specify a value for each parameter before you implement the keyword.</p> <ol style="list-style-type: none"> Click  (Implement with Silk Test) in the Actions column of the keyword that you want to implement. <p> Note: If your browser reports a security warning for opening an external application, confirm the message.</p> <ol style="list-style-type: none"> The Silk Test client opens. Refer to the Silk Test client documentation for detailed information. After having implemented a keyword, upload it back to Silk Central. |
| Editing an implemented keyword in Silk Test | Click  (Open with Silk Test) to open the keyword in the Silk Test client. Refer to the Silk Test client documentation for detailed information. |
| Converting a draft keyword to a keyword sequence (a keyword) | Click  in the Actions column of the keyword that you want to convert to a keyword sequence. The original keyword is then replaced by a keyword sequence with the same name and the keyword editor opens. |

| Task | Steps |
|---|--|
| consisting of other keywords) | |
| Converting a keyword sequence to a draft keyword | <p>If you accidentally created a keyword sequence, you may want to convert it back to a draft keyword. Click  (Convert to draft keyword) in the Actions column of the keyword sequence that you want to convert.</p> <p> Note: Converting a keyword sequence to a draft will delete the sequence and all contained keywords. This action cannot be undone.</p> |
| Editing a keyword sequence | Click  to open the keyword sequence in the keyword editor. |
| Grouping keywords | <p>To better structure the keywords in a library, you can group them:</p> <ul style="list-style-type: none"> • Silk Central keyword sequences can be placed in sub-folders. The folder names are reflected as group names and your keywords are grouped accordingly in the list. • Keyword groups of implemented keywords are defined in the implemented code. These group names are used by Silk Central and your keywords are grouped accordingly in the list. |
| Viewing tests and keyword sequences that reference a specific keyword or keyword sequence | To view all tests and keyword sequences that reference a specific keyword or keyword sequence, select the appropriate entry in the keyword list and the Referenced By list will display all referencing keyword-driven tests and keyword sequences. Simply click the name to jump to the respective reference. |
| Replacing keywords in referencing tests and keyword sequences | <p>To replace references to the selected keyword or keyword sequence with another one, select the appropriate entry in the keyword list and use the Referenced By list to select the objects in which you want to replace the reference.</p> <ol style="list-style-type: none"> 1. Select the keyword or keyword sequence that you want to replace. 2. In the Referenced By list, use Ctrl+Click or Shift+Click to select the objects in which you want to replace the reference. 3. Right-click the selection and choose Replace keyword in selected reference(s). The Replace References dialog box appears. 4. Select the keyword that you want to replace the current keyword with. 5. If the selected keyword contains parameters, click Next, otherwise click OK and you are done. 6. The parameters of the selected keyword are listed. For each parameter, select the matching parameter of the previous keyword, or enter a value. 7. Click OK. |

Keyword Sequence

If your selected object is a keyword sequence (a keyword consisting of other keywords), the **Keywords** page displays the keyword editor, which enables you to manage the keywords of the selected keyword sequence. The following actions are possible:

| Task | Steps |
|---|--|
| Opening a test or keyword sequence in Silk Test | Click Open with Silk Test to open the selected test or keyword sequence in Silk Test. |

| Task | Steps |
|--|--|
| Adding a keyword | <ol style="list-style-type: none"> 1. Click New Keyword at the bottom of the keywords list, or right-click a keyword and select Insert Keyword Above from the context menu. <p> Note: You can let Silk Central recommend keywords based on their usage. Toggle the recommendations on or off with Enable Recommendations or Disable Recommendations in the context menu. For additional information, see <i>Which Keywords Does Silk Central Recommend?</i></p> 2. Select a keyword from the list of available keywords or type a new name to create a new keyword. 3. Click Save. <p>Alternatively, double click an existing keyword in the All Keywords pane on the right or drag and drop it.</p> <p> Tip: You can select multiple keywords with Ctrl+Click. When dropping them, they will be sorted in the order that you selected them in.</p> |
| Deleting a keyword | Click  in the Actions column of the keyword that you want to delete. Click Save . |
| Changing the order of keywords | Drag and drop a keyword to the desired position. Click Save . |
| Creating a keyword sequence (a keyword consisting of other keywords) | <ol style="list-style-type: none"> 1. Select the keywords that you want to combine in the keywords list. Use Ctrl+Click or Shift+Click on the row number column to select multiple keywords. 2. Right-click your selection and click Combine. 3. Specify a Name and Description for the new keyword sequence. |
| Extracting keywords from a keyword sequence | Right-click a keyword sequence and click Extract keywords . The original keyword sequence is then replaced by the keywords that it contained, but it is not removed from the library. Click Save . |
| Defining parameters for a keyword sequence | <ol style="list-style-type: none"> 1. Click Parameters above the keywords list. The Parameters dialog box appears. 2. Click Add Parameter. 3. Specify a Name for the new parameter. If the parameter is an outgoing parameter (delivers a value, instead of requiring an input value), check the Output checkbox. 4. Click OK. 5. Click Save. |
| Editing a draft keyword | 1. Click  in the Actions column of the draft keyword that you want to edit. |

| Task | Steps |
|---------------------------------------|--|
| | <ol style="list-style-type: none"> 2. Select a Group or specify a new group for the keyword. 3. Type a Description for the keyword. This information is valuable for the engineer who will implement the keyword. 4. Click OK. 5. <i>Optional:</i> Click into a parameter field to add parameters for the keyword. If the keyword is implemented with Silk Test, these parameters will appear in the generated code stub. 6. Click Save. |
| Searching for a keyword | <p>Use the search field in the All Keywords pane on the right to find a specific keyword. When you enter alphanumeric characters, the list is dynamically updated with all existing matches. Tips for searching:</p> <ul style="list-style-type: none"> • The search is case-insensitive: <code>doAction</code> will find <code>doaction</code> and <code>DOAction</code>. • Enter only capital letters to perform a so-called <i>CamelCase</i> search: <code>ECD</code> will find <code>Enter Car Details</code>, <code>Enter Contact Details</code> and <code>EnterContactDetails</code>. • Keyword and group names are considered: <code>test</code> will find all keywords that contain <code>test</code> and all keywords in groups where the group name contains <code>test</code>. • <code>?</code> replaces 0-1 characters: <code>user?test</code> will find <code>userTest</code> and <code>usersTest</code>. • <code>*</code> replaces 0-n characters: <code>my*keyword</code> will find <code>myKeyword</code>, <code>myNewKeyword</code> and <code>my_other_keyword</code>. • <code><string>.</code> only searches in group names: <code>group.</code> will find all keywords in groups where the group name contains <code>group</code>. • <code>.<string></code> only searches in keyword names: <code>.keyword</code> will find all keywords that contain <code>keyword</code>. • <code><string>.<string></code> searches for a keyword in a specific group: <code>group.word</code> will find <code>myKeyword</code> in the group <code>myGroup</code>. • Use quotes to search for an exact match: <code>'Keyword'</code> will find <code>Keyword</code> and <code>MyKeyword</code>, but not <code>keyword</code>. |
| Finding draft keywords in the library | <p>If you want to switch to the library that contains the draft keywords of your selected test or keyword sequence, click Draft keywords in the top right of the editor page.</p> |

Building a Keyword-Driven Library Using Java

Build an external keyword-driven library with Java to upload it to Silk Central. This enables you to easily integrate your existing test framework into Silk Central, for example if you already have existing Selenium tests.

1. Select **Help > Tools** and download the **Java Keyword Library Tool**.
2. Add the downloaded `com.borland.silk.keyworddriven.jar` to the classpath of your Java project.
3. Create a Java class that will contain your Keywords.
4. Annotate the Java methods that you want to use as keywords. The following sample outlines a class that contains one keyword:

```
import com.borland.silk.keyworddriven.annotations.Argument;
import com.borland.silk.keyworddriven.annotations.Keyword;

public class Keywords {

    @Keyword("My Keyword")
    public void myKeyword(@Argument("Username") String name) {
        System.out.println(name);
    }
}
```

5. Compile your sources and create a Java archive (jar).
If you are using Java 9 or later, ensure that the `Created-By` property in the `MANIFEST.MF` file contains the Java version that the library was built with.
6. Call the command line tool that is contained in the downloaded `jar` file with the following arguments:
 - `java`
 - `-jar com.borland.silk.keyworddriven.jar`
 - `-build`
 - Library name to be generated.
 - Library type, only Java is currently supported.
 - Binaries path where your keyword library jar file resides and if applicable, any other required libraries.
 - Name of the output library package (zip archive) to be generated.

The following example outlines the command line to build a keyword library:

```
java -jar com.borland.silk.keyworddriven.jar -build
MyLibrary Java ./output ./output/library.zip"
```

Select **Tests > Libraries > <Library> > Properties** . The generated zip archive can be uploaded to Silk Central with the command line tool or in the **Keyword Library** section.

Keyboard Functions (Libraries)

The following actions can be performed on the selected nodes:

| Keys | Action |
|--------------|---------------------|
| Up | Move selection up |
| Down | Move selection down |
| Right | Expand element |
| Left | Collapse element |

| Keys | Action |
|--------------------|--|
| F2 | Edit element |
| Ctrl+C | Copy element |
| Ctrl+X | Cut element |
| Ctrl+V | Paste element |
| Del | Delete element |
| Click | Select an element and remember it as the current element. |
| Ctrl+Click | Toggle the selection status of the clicked element and add/remove additional elements with further clicks. |
| Shift+Click | Select the span from the currently-selected element to a newly selected element. |
| Shift+Up | Select element above. |
| Shift+Down | Select element below. |

Managing Shared Objects

A shared object is an object that is included in a library and that can be used by multiple tests in multiple projects.

The following object types can be shared:

- Business components. These can be used for business process testing.
- Flows. These can be used for business process testing.
- Shared steps. These can be used in manual tests or in business components.
- Keywords and keyword sequences. These can be used for keyword-driven testing.

Calling Shared Steps

To reduce the effort for maintaining the same test steps in various tests and projects, you can add shared steps from a library as a single step to a test or shared object.

You can only add a call to all steps in a shared object, not to a selection of them. When a call to a shared object is added, all shared steps are displayed as a single step in the calling test or shared object and you can add additional steps before and after the called steps. The initial shared steps are only displayed during execution of the referencing test. The shared object cannot be deleted. To delete the shared object, you first have to remove all calls to it.

The name of the shared object, whose steps are called, is displayed as the name of the step in the **Steps** page. Click on the name to access the called shared object.

Adding a Call to a Shared Steps Object

To add a call to a shared steps object:

1. In the menu, click **Tests > Details View** to add the shared steps to a test or click **Tests > Libraries** to add the shared steps to another shared steps object.
2. In the corresponding tree, click on the node to which you want to add the shared steps.
3. Click the **Steps** tab.
4. To add the shared steps to the end of the steps list, click . To insert it above the selected step, click . The **Call to Shared Steps** dialog box opens.
5. Select the shared steps node, whose steps you want to add, from the tree.



Note: For a test, the tree displays only the libraries that are visible to the active project. For more information, see the *Visibility Page*.

6. Click **Ok**.

Call to Shared Steps Dialog Box

The **Call to Shared Steps** dialog box enables you to browse shared steps objects in the **Libraries** tree. To open the dialog box, choose a manual test in the **Tests** tree or a shared steps object in the **Libraries** tree, click the **Steps** tab and click .

To search for existing libraries, folders, and shared steps objects in the **Libraries** tree, type the name you are searching for in the **Find** field of the dialog box and press **Enter**. The search is case-insensitive and not limited to entire words. Libraries and folders returned by the search are displayed with their entire subtree.

Creating a Shared Steps Object

You can create a shared steps object in the **Steps** page of any manual test or shared object.

To create a shared steps object:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Navigate to the **Steps** page of the manual test or shared object that includes the steps you want to share.
3. Select the steps with **Ctrl+Click** or **Shift+Click**.
4. Click . The **Create Shared Steps** dialog box opens.
5. Enter a name for the new shared steps object in the **Name** field.
6. Choose the location where you want to place the new shared steps object in the **Libraries** tree.
7. Click **OK**. The shared steps object is created and the selected steps are replaced by a call to the shared steps object.

Extracting Shared Steps from Libraries

To make a manual test or a shared steps object independent of other shared steps objects, extract the steps from the libraries that contain the shared steps objects.

All called steps are then copied into the manual test or shared steps object.

To extract the shared steps included in a manual test or a shared steps object:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. In the corresponding tree, click on the node in which you want to resolve the shared steps.
3. Click the **Steps** tab.
4. Select the steps that you want to extract with **Ctrl+Click** or **Shift+Click**.
5. Click  and confirm the message box.

Managing Shared Step Properties

Tests > Test Step Properties

Tests > Libraries > <Library> > Step Properties

To create, edit, or view step properties for shared step objects in the **Libraries** tree, select the library and click the **Step Properties** tab. The **Step Properties** page lists all properties that can be populated into manual test steps across the active project. To create a new step property, click **New Property**. The actions that can be performed on a property are **Delete**, **Move Up**, and **Move Down**.

Adding or Editing Manual Test Steps

Add or edit manual test steps of a test or a shared steps object in the **Steps** page.

To add or edit a manual test step:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the node for which you want to add or edit steps in the **Tests** or **Libraries** tree.
3. Click the **Steps** tab.
4. Click on the test step that you want to edit, or use the icons on the **Steps** toolbar to insert a new step. The details of the test step are shown in the **Edit Step** view.
5. Edit the name, action description, expected results, and step properties of the test step in the corresponding fields.

You can insert values from data sources into manual test steps in the form of parameters (see *Adding a Data Source Value to a Manual Test Step*). Parameters use the syntax `${<name>}`, for example `${username}`. When converting a manual test to a keyword-driven test, these parameters are automatically added to the generated keyword.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

6. Perform one of the following actions to finish editing the test step:
 - Click **Save and New Step** to save your changes and add a new step after the current one.
 - Click **Save** to save your changes.
 - Click **Discard** to discard your changes.

Exporting and Updating Shared Steps Objects

Use the export-update to Excel functionality if you want to perform bulk edits on shared steps objects.

Export and Update Overview

The following is a high level overview for exporting and updating:

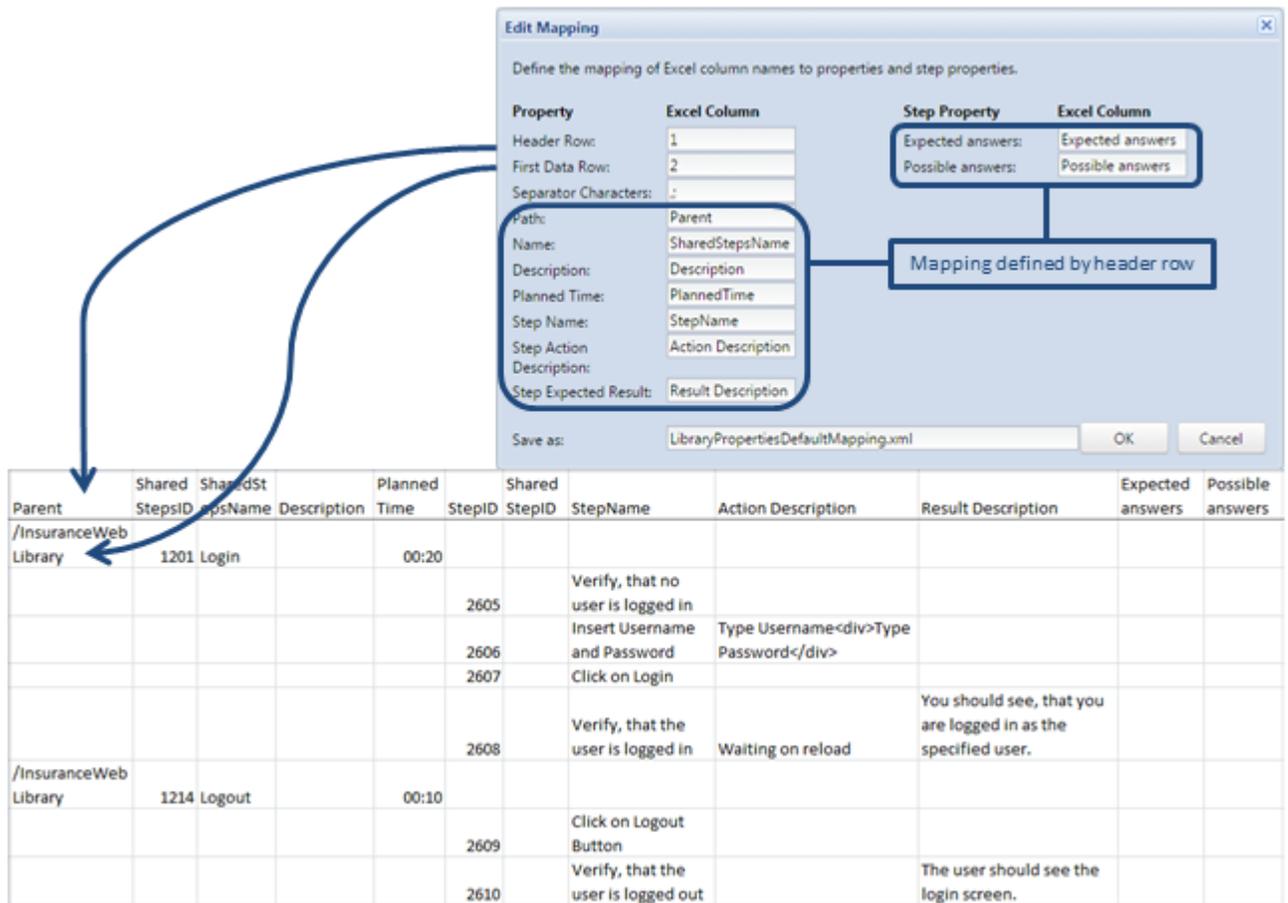
- Select the library.
- Export to Excel.
- Make changes in Excel.
- Verify/create the mapping file.
- Update back into Silk Central.

Exporting Shared Steps Objects to Excel

Export shared steps objects to Excel to perform bulk edits.

1. In the menu, click **Tests > Libraries**.
2. Right click the library that includes the shared steps objects that you want to export and select **Export to Excel**. The **Export Library to Excel File** dialog box appears.
3. Select the mapping file from the **Choose Mapping File** list box.
The default mapping file is `LibraryPropertiesDefaultMapping.xml`.
4. Click **Edit Mapping** to edit the mapping file. The **Edit Mapping** dialog box appears.
5. Specify the Excel columns that should be mapped to the properties and step properties (if step properties exist).

The following example explains the mapping:



- Optionally, change the name of the mapping file in the **Save as** field. Click **OK**. The **Export Library to Excel File** dialog box appears.
- Click **OK**. The Excel file is created and downloaded in your browser.

Open the Excel file and edit your shared steps objects. Save the file when you are done to upload it back into Silk Central.

Restrictions: If you create, delete, or move shared steps objects in the Excel file, these changes will be ignored when updating them back into Silk Central.

Updating Shared Steps Objects from Excel

After updating your shared steps objects in Excel, update them back into Silk Central.

- In the menu, click **Tests > Libraries**.
- Right click the library that includes the shared steps objects that you want to update and select **Update from Excel...**. The **Update Library from Excel File** dialog box appears.
- Select the mapping file from the **Choose Mapping File** list box.
The default mapping file is `LibraryPropertiesDefaultMapping.xml`.
- Optional:* Click **Edit Mapping** to edit the mapping file. This is only necessary if the mapping file that was used for exporting the shared steps objects was modified, or if the columns headers were modified in the Excel file. For detailed information on mapping properties and step properties, see [Exporting Shared Steps Objects to Excel](#).
- Click **Choose File** to select the Excel file in the **Library File (.xlsx)** field.
- Click **OK**. Your changes are updated in Silk Central.

Steps Page

Tests > Details View > <Manual Test> > Steps

Tests > Libraries > <Shared Steps Object> > Steps

Tests > Libraries > <Business Component> > Steps

The **Steps** page lists all manual test steps that are included in the selected test, shared steps object, or business component. The page supports standard Windows Explorer style multi-select functionality.

To create a new shared steps object, select the steps you want to share and click .

The page includes the following toolbar items:

| Toolbar Item | Description | | | | | | | | |
|---|---|--------------------|---|--------------------|--|---------------------------|--|---------------|--------------------------|
|  | Add a new test step to the end of the list. | | | | | | | | |
|  | Insert a new test step before the selected step. | | | | | | | | |
|  | Add a call to a shared steps object from a library to the end of the list. | | | | | | | | |
|  | Add a call to a shared steps object from a library before the selected step. | | | | | | | | |
|  | Delete the selected test steps from the list. | | | | | | | | |
|  | Cut the selected test step from the list and move it to the clipboard. | | | | | | | | |
|  | Copy the selected test steps to the clipboard. | | | | | | | | |
|  | Paste a copy of the test steps held on the clipboard to the row above the selected in the list. | | | | | | | | |
|  | Move the selected test step one position up in the list. | | | | | | | | |
|  | Moves the selected test step one position down in the list. | | | | | | | | |
|  | Opens the Attachments dialog box, where you can perform the following actions: <table border="0" data-bbox="324 1291 1218 1470"> <tr> <td>Upload File</td> <td>Upload and attach a file to the selected test step.</td> </tr> <tr> <td>Attach Link</td> <td>Attach a link to the selected test step.</td> </tr> <tr> <td>Download All Files</td> <td>Download all attached files from the selected test step.</td> </tr> <tr> <td>Delete</td> <td>Delete the file or link.</td> </tr> </table> | Upload File | Upload and attach a file to the selected test step. | Attach Link | Attach a link to the selected test step. | Download All Files | Download all attached files from the selected test step. | Delete | Delete the file or link. |
| Upload File | Upload and attach a file to the selected test step. | | | | | | | | |
| Attach Link | Attach a link to the selected test step. | | | | | | | | |
| Download All Files | Download all attached files from the selected test step. | | | | | | | | |
| Delete | Delete the file or link. | | | | | | | | |
|  | When viewing a called shared steps object, click here to view the original steps. The resolved steps are marked with an arrow, and you can access them by clicking on their name. You cannot edit steps while this view is enabled. | | | | | | | | |
|  | View steps called from a shared steps object in a library as a single step. Enable this view to edit test steps. | | | | | | | | |
|  | Create a shared steps object out of the selected steps. | | | | | | | | |
|  | Extract calls from other shared steps objects. Copies all steps in the calls you have selected in the grid into the manual test, shared steps object, or business component. | | | | | | | | |

The **Steps** page supports the following keyboard functions (shortcuts) for test elements:

| Keys | Action |
|---------------|------------|
| - | Move up |
| + | Move down |
| Ctrl+A | Select all |
| Ctrl+C | Copy |
| Ctrl+X | Cut |
| Ctrl+V | Paste |
| Ins | Insert |
| Del | Delete |

The steps are displayed in a grid with the following columns:

| Column | Description |
|-----------------------|--|
| Order | Number of the step in the execution sequence. |
| Name | Name of the test step. |
| Action Description | Action you must perform to execute the test step. |
| Expected Results | Expected result of the test step. |
| Attachments | Amount of links or files that are attached to the test step. |

The **Edit Step** view shows the details of the selected test step and allows you to edit the name, action description, expected results, and step properties of the test step.

The following buttons are available in the **Edit Step** view:

| Button | Description |
|--------------------------|---|
| Save and New Step | Click when you are finished with editing the current step and you wish to save your changes and add a new step after the current. |
| Save | Click when you are finished with editing the current step and you wish to save your changes. |
| Discard | Click to discard your changes. |

Versions

A version is a revision of an element in a certain state.

The following elements have versions:

- Tests.



Note: Data sets and inherited test properties, parameters, and attributes are not included in test versions.

- Shared objects.

You can manually create versions or revert to previous versions. All versions of an element are listed on the **History** page of the element. When you copy a project, the versions are also copied. If you are working with shared tests, see also [Maintaining Versions of Shared Tests](#).

Creating a Version

Create a new version of an element to track or to revert any changes to the element.

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click  in the toolbar.
This action is only enabled if the shared object or the test was changed.
The **Create Version** dialog box appears.
4. *Optional:* Enter a comment on the version in the **Comment** field.
5. Click **Ok**. The new version is shown in the **History** page of the element.

Viewing a Version

To see if a specific version of a shared object or a test applies to your testing needs, view the details of the version.

To view the details of a version of a shared object or a test:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click the **History** tab.
4. Right click on the version of which you want to see the details.
5. Click **View version <ID>**.
6. The **View Version** dialog box opens. If the version includes attachments, you can view the attachments by clicking on their number in the **Attachments** column of the **Steps** section.



Note: Use the arrows to change the version number and view additional versions.

Comparing Versions

To view the differences between two versions of a shared object or a test, compare the versions.

To compare two versions of a shared object or a test:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click the **History** tab.
4. Select the two versions you want to compare with **Ctrl+Click**.
5. Right click on the selection.
6. Select **Compare Version <x> and <y>**. The **Compare Versions** dialog box opens, displaying all the added, changed, and removed content.



Note: Use the arrows to change the version numbers and compare additional versions.

Reverting to a Previous Version

To undo changes to a shared object or a test, revert to a previous version. Note the following special cases when reverting to a previous version:

- If the value of an inherited attribute has changed compared to the current version, the most current value is again inherited in the reverted version.

- If the type of an attribute has changed compared to the current version, the attribute is ignored in the reverted version.
- If the value of a list-type attribute no longer exists in the current version, the original value is added to the attribute again and used in the reverted version.

To revert a shared object or a test to a previous version:

1. For a shared object, click **Tests > Libraries** in the menu. For a test, click **Tests > Details View** in the menu.
2. Select the shared object in the **Libraries** tree, or the test in the **Tests** tree.
3. Click the **History** tab.
4. Right-click on the version to which you want to revert the shared object or test to.
5. Click **Revert to Version <ID>**.
6. The shared object or test is reverted to the selected version and a new version is created in the **History** page. Attachments and data sets are not reverted for tests.

Library History Page

The **History** page displays the history of the selected shared object.

The **History** page includes an entry for each version of the selected shared object, and allows you to create new versions, view version details, revert to a previous version, and view the differences between versions.

To display only versions in the **History** page, and no other entries, open the **Version** column header's context menu, select **Filter** and type 0 in the > field.

The page shows the following items for each listed entry:

| Item | Description |
|------------|--|
| Actions | Only for version entries. The actions you can perform on a selected version are View Version , Delete Version and Revert to Version . |
| Version | Only for version entries. The version of the shared object. |
| Comment | A comment on the changes. |
| Created On | Date and time this history entry was created. |
| Created By | Name of the user who created this history entry. |

Associating Requirements with Tests

Associate your requirements and tests to enable measuring how well your software covers the specified requirements.

Assigning Requirements to Tests

To manually assign requirements to tests:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select the test to which you want to assign requirements.
3. Click the **Assigned Requirements** tab.

 **Note:** If you have created a requirements filter, you can select it from the filter list above the **Requirements** tree. To create a new requirements filter, click **Requirements > Details View** in the menu and click  on the toolbar.

All requirements that are available for assignment are displayed in the **Requirements** tree.

4. Select the requirements that you want to assign.

You can use your keyboard's **Ctrl** and **Shift** keys to select multiple requirements using standard browser multi-select functions.

5. Click **Assign All Filtered** to assign all filtered requirements to the selected test, or drag your selection and drop it into the **Assigned Requirements** area, or click  to the left of any requirement. Assigning all filtered requirements is only enabled if a filter is selected and if you have either **Manage requirements** or **Manage assets** permission.

 **Note:** Newly generated tests can automatically be assigned to the requirements from which they are generated by checking the **Assign newly generated tests to requirements** check box on the **Generate Tests from Requirements** dialog box. This is the default behavior.

Sorting Assigned Requirements

To sort the assigned requirements:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a test.
3. Click the **Assigned Requirements** tab.
4. Click the column header of the property by which you want to sort the requirements.

A small upward or downward pointing arrow indicates both which column the sort is based and the direction of the sort, ascending or descending.

5. If required, click the column header again to reverse the direction of the sort.

Locating Assigned Requirements

To locate assigned requirements in the **Available Requirements** tree:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a test.
3. Click the **Assigned Requirements** tab.
4. In the **Actions** column of a requirement, click  to find out in which node in the **Available Requirements** tree the requirement is stored in.

The corresponding parent-requirement node is expanded and the assigned requirement is highlighted.

Removing Requirement Assignments

To remove a requirement assignment:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a test that has at least one requirement assigned to it.
3. Click the **Assigned Requirements** tab.
4. In the **Actions** column of the assigned requirement, click , or use **Ctrl+Click** or **Shift+Click** to select multiple requirements and press **Del**.

 **Note:** To remove all requirement assignments from the selected test, click **Remove All**.

Test Assigned Requirements Page

Tests > Details View > <Test Element> > Assigned Requirements

The **Assigned Requirements** page lists the requirements that have been assigned to the selected test or project, and allows you to assign additional requirements. The **Available Requirements** tree lists all requirements that can be assigned.



Note: If you have created a requirements filter, you can select it from the filter list above the **Requirements** tree. To create a new requirements filter, click **Requirements > Details View** in the menu and click  on the toolbar.

For each assigned requirement, the page displays the following columns:

| Column | Description |
|-------------|---|
| Actions | Actions that you can perform on the selected requirement. Remove Requirement , Locate , and View Description . |
| Requirement | Name of the assigned requirement. Click to open the Requirements > Properties page. |
| Priority | Priority of the requirement. |
| Risk | Potential risk associated with the requirement. |
| Reviewed | Review status of the requirement. |

Test Reports

This section explains the test-related reports that ship with Silk Central. Test reports give you an overview of the progress of your tests and the status of defects over a period of time or over a range of builds.

Status Reports

The following status reports are available for the **Tests** area:

| | |
|--|--|
| Tests per Component | Represents an overview of coverage of components by test; makes it easier to see where testing activity is needed. |
| Implemented Tests (per component) | Represents an overview of coverage of components by tests that have the <i>Implemented</i> attribute set to <i>Yes</i> . |
| Failed Tests (per component) | Represents an overview of failed tests per component; makes it easier to identify the most critical components in the environment. |
| Tests (per test container) | Represents a success rate for each test container by listing the number of passed tests. |
| Quality Goals Planning Report | Represents the number of tests in each goal criteria along with the planned time rolled up to the goal criteria. |



Note: The planned time value is only for manual tests. Automated tests do not have planned time.

This report is split three ways for each quality goal defined in a project:

- **Testing Scope** - compares the total number of tests available against the total number or required tests for each item in a quality goal.
- **Required Planned Time** - compares the required planned time among the items in the quality goals.

- **Value table** - lists all of the quality goal items along with related metrics: **Tests Available**, **Goal (%)**, **Tests Required**, and **Planned Time for Manual Tests**.



Note: For a multi select attribute/property, the test will be counted for each matching occurrence.

Quality Goals Execution Report

Visually displays the number of tests executed for each item of each goal. This report is split into two sections for each quality goal defined in a project:

- **Execution Status Graph** - lists each quality goal item with a color-coded execution status and summary value.
- **Value table** - lists all of the quality goal items as well as their execution metrics: **Passed**, **Failed**, **Not Executed**, **N/A**, **Total**, **Tests to Meet Goal**, and **Execution % to Goal**.

Test Status Overview

Represents a status overview of all tests, structured by the statuses *Passed*, *Failed*, *Not Executed*, and *N/A*.

Test Status Overview (per test container)

Represents a status overview of all tests contained in a specific test container, structured by the statuses *Passed*, *Failed*, *Not Executed*, and *N/A*.

Progress Reports

The following progress reports are available for the **Tests** area:

Test Progress Across Builds 'X' and 'Y'

Represents a trend in test progress resulting from viewing test statuses in context with builds. The user must specify a build range consisting of a start- and an end-build.

Manual Test Coverage since Build <x> (cumulative)

Represents a trend in manual test progress, excluding all kind of automated tests, resulting from viewing test statuses in context with builds. The user must specify a starting build number which is used as basis for calculating a cumulative progress.

Automated Test Coverage for Builds (non-cumulative)

Represents the automated test coverage resulting from viewing test statuses in context with builds. The user must specify a version and a product and the report shows the test coverage that could be achieved with the automated tests per build.

Progress of Tests in Specific Folder Over the Past <x> Days

Represents a trend in test progress resulting from viewing test statuses for the test tree node with the given ID and for the last <x> number of days.

Test Progress Over the Past 'X' Days

Represents a trend in test progress by considering test statuses over the past 'X' days.

Test Progress this Month

Represents a trend in test progress resulting from viewing test statuses for the current month.

Percentage Testing Success Over the Past 'X' Days (per component)

Represents a percentage listing of successful tests per component over the past 'X' days; assists in identifying the components in the environment that are most critical.

Tests Created in the Past 'X' Days (per component)

Represents a listing of new tests per component over the past 'X' days. Assists in identifying components that lack testing activity.

Manual Test Reports

The following manual-test reports are available for the **Tests** area:

| | |
|--|--|
| Historic Planned vs. Actual Execution Time (per user) | Represents an overview of planned and actually required execution time for completed manual tests per user over a specific period of time. |
| Planned vs. Actual Execution Time (status per user) | Represents progress in terms of planned vs. actual hours of currently pending manual tests per user. Manual tests are only considered if test results have been entered by the user and are assigned to the user who enters the results. |
| Manual Test Result Document | An easily printable manual test case report for the latest results of all tests assigned to the specified execution plan. |
| Manual Test Result Document (Microsoft Word) | An easily printable manual test case report for the latest results of all tests assigned to the specified execution plan in a Microsoft Word document. |
| Manual Test Results by Execution Plan | An easily printable manual test case report for the latest results of all tests assigned to the specified execution plan folder or testing cycle. |
| Planned vs. Actual Execution Time of Manual Tests (Summary) | Represents an overview of the deviation between planned and actual time for execution of manual tests, viewed on a daily basis. |

Baseline Tests Comparison Report

The **Baseline Tests Comparison** report compares a baselined project with the baseline and displays the number of changed, deleted, and created tests.

The input parameters for the **Baseline Tests Comparison** report are the identifiers of the project and the baseline.

Test Run Comparison Report

Test run-comparison reports compare the last two runs of a test by providing detailed information about context, status, errors, warnings, duration, properties, attributes, parameters, and success conditions.



Note: Run-comparison reports are not suitable for the comparison of manual tests to automated tests. When the name of the report includes (*Last Two Runs*), you can compare only the last two runs of the execution plan or test.

Assigned Execution Plans

Assign tests to execution plans to maintain control over test executions during development and testing.

Viewing Assigned Executions

To view the execution plans that the selected test is assigned to:

1. In the menu, click **Tests > Details View** .
2. Select the test for which you want to view the execution plan assignments.
3. Click the **Properties** tab. In the **Executions** section, the complete list of execution plans that the test is assigned to is displayed in the **Assigned Execution Plans** grid.

Test Views

Silk Central offers several views and tools to manage your tests.

Working with the Tests Tree

Tests > Details View

As with requirements, tests are displayed, organized, and maintained through a hierarchical tree structure, the **Tests** tree. The **Tests** tree enables you to organize tests in any number of hierarchy levels. Each node in the tree represents either a test, a test folder, or a test container.



Note: When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

Keyboard Functions (Tests)

The following actions can be performed on the selected nodes:

| Keys | Action |
|--------------------|--|
| Up | Move selection up |
| Down | Move selection down |
| Right | Expand element |
| Left | Collapse element |
| F2 | Edit element |
| Ctrl+C | Copy element |
| Ctrl+X | Cut element |
| Ctrl+V | Paste element |
| Del | Delete element |
| Ctrl+Up | Move element up |
| Ctrl+Down | Move element down |
| Click | Select an element and remember it as the current element. |
| Ctrl+Click | Toggle the selection status of the clicked element and add/remove additional elements with further clicks. |
| Shift+Click | Select the span from the currently-selected element to a newly selected element. |
| Shift+Up | Select element above. |
| Shift+Down | Select element below. |
| Enter | Assign element (only in assignment tree) |

Expanding or Collapsing the Tests Tree

You can consolidate levels of the **Tests** tree or display all levels of the tree based on your viewing needs.

To collapse or expand levels of the **Tests** tree:

1. In the menu, click **Tests > Details View** .
2. Select the test container or test folder in the **Tests** tree and continue with one of the following options:
 - Click ▶ to the left of the name of the test element to expand the element.
 - Click ◀ to the left of the name of the test element to collapse the element.
 - Right-click on the element and select **Expand** or **Collapse**.

Cutting, Copying, Pasting, and Deleting Test Elements

The toolbar in the **Tests** area enables you to easily cut, copy, paste, and delete test elements within the **Tests** tree, both within the current project and between projects. These editing functions simplify the process of building and managing the tests of your project.

The data types that are copied along with tests and test folders are properties, attributes, parameters, and attachments. Assignments, issues, runs, and history are not copied.

 **Note:** Containers cannot be copied or pasted.

To cut, copy, paste, or delete test elements:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select the elements that you wish to edit. You can select multiple elements with **Ctrl+Click** or **Shift+Click**.
3. Click the appropriate toolbar button:

Button Description



Deletes the selected elements from the tree.



Cuts the selected elements from the tree and moves them to the clipboard.



Copies the selected elements to the clipboard.



Pastes a copy of the elements held on the clipboard to the same level of the currently selected element.



Pastes a copy of the elements held on the clipboard as a sub-node of the currently selected element.



Note: The commands are also available through the context menu in the **Tests** tree and with [keyboard shortcuts](#).

Copying and Pasting a Test Element from One Project to Another

To copy and paste a test folder or test between projects:

1. In the menu, click **Projects > Project List**.
2. Select the source project.
3. In the menu, click **Tests > Details View** .
4. Click  to cut or  to copy the element to the clipboard.
5. In the menu, click **Projects > Project List**.
6. Select the destination project.
7. Select the destination container and folder.
8. Click .



Note: The commands are also available through the context menu in the **Tests** tree and with [keyboard shortcuts](#).

Sharing and Consuming Tests

Manage recurring tests and their requirement assignments centrally and share and consume these tests across projects.

If your project is configured to consume tests from a specific project that is sharing tests (see [Adding Projects](#)), you can reuse these tests into your project by selecting them through the selection UI.

To insert shared tests into your project:

1. In the menu, click **Tests > Details View**.
2. Right-click on the test container or the test folder into which you want to insert the shared tests.
3. Select **Shared Tests**.
4. Select **Insert Shared Tests**. The **Browse Tests** dialog appears.
5. Select the tests that you want to insert.



Tip: To narrow down the displayed test set, either set some criteria in the **Search** field or select a pre-defined filter from the list.

6. Click **OK**. The selected tests are inserted. The test properties contain a **Shared Test Reference**, which means a link to the shared test.
7. To detach a consuming test from a shared test, click **X** in the **Shared Test Reference** field.



Note: To remove a reference to a shared test, you require the **Tests and Executions - Manage assets** permission.

After removing the reference to a shared test, the test will no longer have any connection to the shared test. Updates will therefore no longer be pushed and received between the shared test and the consuming test.

To see all tests that consume a specific shared test, click **Tests > Details View** in the menu and select the **Usages** tab.

When consuming tests, consider the following:

- The project that consumes the tests must be configured to reuse tests from the project that is sharing the tests.
- You can only insert tests into an existing test container.
- The actions **Accept Changes** and **Push Changes to Shared Test** handle differing test types between a consuming test and the corresponding shared test, by resulting in the same test type being used in both tests.
- When accepting changes from a test which uses shared steps, keywords, business components, or flows, the corresponding libraries must be visible in both the project that consumes the test and the project that shares the test.
- To retain assigned requirements, the project that consumes a test must either use the same requirements integration or the same Word or Excel import file as the project that is sharing the test.
- You can only update a requirement from Excel for which a unique external ID is defined in the Excel import file. Silk Central requires this ID to identify which requirement needs to be updated.
- Data-driven children, test package children, and linked test containers cannot be shared.
- Inherited test properties, parameters, and attributes are not considered in consuming tests. Such tests will inherit what is inherited in the consuming project. Accepting changes from the original test can lead to properties, parameters, and attributes being removed in consuming tests if inheritance is being used, as these properties, parameters, and attributes might not be available in the consuming project.
- Assigned issues are not copied to consuming tests.
- If you use test attributes, they must have the same name and type in both projects.
- When copying or baselining a consuming project, the references to the project that is sharing tests are retained. However, these references are deleted when exporting a consuming project.
- The bulk export/update interface and the web services do not yet consider links to original assets.
- You can also insert folders, including their complete hierarchy.

Sharing Tests Out of a Consuming Project

Share tests out of a consuming project.

When you have extended the test set in a project that consumes tests from a project sharing tests, you can make these new tests available to other consuming projects by adding the tests to the shared test set in the sharing project.

1. In the menu, click **Projects > Project List**.
2. Select the consuming project that contains the new tests.
3. In the menu, click **Tests > Details View**.
4. Right-click on the test folder or the test that you want to share.
5. Select **Shared Tests > Share Tests**.



Note: This action is only available if the selection does not contain any tests that are already shared.

6. Select the target folder within the sharing project, where you want to insert the selected folders or tests.
7. Click **OK**. The selected test folders or tests are copied as a child to the selected folder in the sharing project.

Reordering Test Elements

To reorder a test, test folder, or test container:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select the test element that you wish to reorder.
3. Click  to move the test element up one step or click  to move the test element down one step.

Setting a Test Node as Integration Default for External Agile Planning Tools

To use the Web service calls to create tests in Silk Central through an external agile planning tool, set a folder or container in the **Tests** tree as the integration default node, where the Web service will create the test. If you do not specify the integration default node, an error message box displays.

To specify the integration default node in the **Tests** tree:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select the folder or container which you want to set as the integration default node.
3. Right-click the folder or container and select **Set as Integration Default**.



Note: If an integration default node already exists, the default node is changed to the new node.

The integration default node is set to the selected node, enabling the agile planning tool to create tests at this location.



Note: The integration default node is shown in the **Properties** page of the project, in which the node is located.

Filtering on a Folder or Container

To filter on a folder or container:

1. In the menu, click **Tests > Details View**.
2. Right-click the folder or container in the tree that you want to filter and select **Filter Subtree**. The tree displays only the contents of the selected folder or container and the hierarchy up to the root node of the tree.



Note: To remove filtering and display all elements, select **<No Filter>** from the **Filter** list box on the toolbar. Empty folders are not shown in the filtered sub-tree.

Finding Test Properties

The **Find** command in the **Tests** area enables you to locate test property values that meet specified search criteria. Use the **Next**, **Previous**, **First**, and **Last** functions to step through the results of a search for a specified property value. **Find** is enabled across all Silk Central plug-ins and functional categories.



Note: Data-driven test property values can be found using the **Find** command.



Note: When the **Tests** tree is constrained by a filter, **Find** is only executed against those tests that are presented in the **Tests** tree after filtering.

To find a specific test property value:

1. In the menu, click **Tests > Details View** .
2. Click  on the toolbar. The **Find** dialog box opens.
3. From the **Category** list box, select the functional category or Silk Central plug-in across which you want to search.



Tip: When you define a custom plug-in property, it is automatically added to the list.

4. From the **Find in** list box, select the property within which the query should search for the value. The properties available in this list vary based on the selected category.
5. In the **Find what** field, type the alphanumeric string the query should search for.

Optional settings are available for qualifying the query further. Check the check boxes of those that are appropriate:

6. Choose the location in the **Tests** tree where the search should begin.

The following options are available:

| Option Button | Description |
|-----------------------------|---|
| Start from selection | Specifies that the search begins from the currently selected test element. |
| Start from top | Specifies that the search begins from the root of the Tests tree. |
| Find in subtree only | Specifies that the search runs only in the sub-tree which has the currently selected element of the Tests tree as its root node. |

7. Check the appropriate check boxes to further qualify the query.

The following check boxes are available:

| Check Box | Description |
|---------------------------------|--|
| Case sensitive | Specifies that the string is searched case-sensitively.  Note: When using a case sensitive SQL Server, case-insensitive searching is not supported for the following fields: <ul style="list-style-type: none">• Test description• Manual step description• Manual step action description• Manual step expected results |
| Match whole word only | Specifies that search results only include complete standalone instances of the query string. |
| Include read-only values | Specifies that search results include text strings that cannot be directly edited because they are inherited from another test, referenced from a linked test container, or called from a data source in the course of data-driven testing. |

8. Click **Find** to begin the search and advance to the first test container, test folder, or test returned by the query.
9. If your query returns multiple test elements, you are presented with options to advance through the elements.

 **Note:** The **Find** command allows you to search test elements where the search string is an inherited value.

Replacing Test Properties

The **Replace** command enables you to replace identified property values with new values. **Replace** is enabled across all Silk Central plug-ins and functional categories.

 **Note:** Data-driven test property values cannot be replaced using the **Replace** command.

 **Note:** When the **Tests** tree is constrained by a filter, **Replace** is only executed against those tests that are presented in the **Tests** tree after filtering.

To replace an identified test property value:

1. In the menu, click **Tests > Details View**.
2. Click  on the toolbar.
The **Replace** dialog box opens.
3. From the **Category** list box, select the functional category or Silk Central plug-in across which you want to search.

 **Tip:** When you define a custom plug-in property, it is automatically added to the list.

4. From the **Find in** list box, select the property within which the query should search for the value.
The properties available in this list vary based on the selected category.
5. In the **Find what** field, type the alphanumeric string the query should search for.
Optional settings are available for qualifying the query further. Check the check boxes of those that are appropriate.
6. In the **Replace with** field, type the alphanumeric string with which you want to replace the found values.
7. Click **Find** to begin the search and advance to the first test container, test folder, or test returned by the query. Or click **Replace all** to replace all instances of the queried string with the replacement string.
8. If your query returns multiple test elements, you are presented with options to advance through the elements.

 **Note:** The **Replace** command does not allow you to search test elements where the search string is an inherited value.

Test Contents Page

Tests > Details View > Contents

The **Contents** page displays the child elements of the selected project, test container, or folder in the **Tests** tree. The page supports Standard Windows Explorer style multi-select functionality.

 **Tip:** To drill down into the selected folder or container, press **Enter** or double-click the selected item.
Press **Backspace** or click  on the toolbar to navigate one level up.

 **Note:** You cannot copy or paste a test container.

 **Tip:** As with test elements listed in the **Tests** tree, you can right-click the elements listed on the **Contents** page to access context-relevant commands through a context menu. Commands that are

not available are grayed out. Before you can paste a test element into the **Contents** page you must explicitly select an element within the page to gain the application's focus.

 **Note:** When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

The **Contents** page supports the following keyboard functions (shortcuts) for test elements:

| Keys | Action |
|--------------------|--|
| Up | Move selection up |
| Down | Move selection down |
| Right | Deselect |
| Left | Deselect |
| F2 | Edit element |
| Ctrl+A | Select all |
| Ctrl+C | Copy element |
| Ctrl+X | Cut element |
| Ctrl+V | Paste element |
| Del | Delete element |
| Ctrl+Up | Move element up |
| Ctrl+Down | Move element down |
| Click | Select an element and remember it as the current element. |
| Ctrl+Click | Toggle the selection status of the clicked element and add/remove additional elements with further clicks. |
| Shift+Click | Select the span from the currently-selected element to a newly selected element. |
| Shift+Up | Extend selection up |
| Shift+Down | Extend selection down |
| Pos1 | Select first item |
| Shift+Pos1 | Select up to first item |
| End | Select last item |
| Shift+End | Select down to last item |

Test Toolbar Functions

Tests > Details View

The test toolbar provides important commands for managing your tests.

 **Note:** The commands in the test toolbar are also available through context menus in the **Tests** tree.

The following commands are included in the test toolbar:

| Command | Icon | Description |
|----------------------|---|--|
| Document View |  | Displays the Document View , which shows selected properties of all test elements in a single view. |

| Command | Icon | Description |
|-----------------------|------|--|
| Details View | | Displays the Details View , which enables you to drill deeply into the properties of a single test. |
| Grid View | | Displays the Grid View , which shows all tests in a grid. |
| Up | | Navigates one level up in the hierarchy of the navigation tree, regardless of the current cursor focus. |
| New Test Container | | Enables creation of new test containers. |
| New Child Test Folder | | Enables creation of new test folders as sub-nodes of the selected folder or container. |
| New Child Test | | Enables creation of new tests as sub-nodes of the selected folder or container. |
| Edit | | Edit the selected test element. |
| Create Version | | Create a new version of the selected manual test. |
| Delete | | Delete the selected test element. |
| Cut | | Cut a test element from the Tests tree and save it to the clipboard. |
| Copy | | Copy a test element from the Tests tree to the clipboard. |
| Paste | | Paste a test element from the clipboard to the Tests tree. |
| Paste as Child | | Paste a test element from the clipboard as a child element to the currently selected test element. |
| Move Up | | Move a test element up within the Tests tree. |
| Move Down | | Move a test element down within the Tests tree. |
| Find | | Search through all test elements in the active project based on configurable parameters. |
| Replace | | Replace instances of found values with a new value. |
| New Filter | | Create a new custom filter for the Tests tree. |
| Filters | | Lists the available filters for the Tests tree |
| Edit Filter | | Edit the currently selected custom filter. |
| Delete Filter | | Delete the currently selected custom filter. |
| Copy Filter | | Copy the currently selected custom filter. |
| Show Changes | | Show recent changes to the tests. |
| Acknowledge | | Acknowledge changes to the tests. |
| Download as XLSX | | This function is enabled in the Document View and Grid View . In the Document View , the action generates an Excel sheet that includes all currently visible rows. In the Grid View , the action generates an Excel sheet that includes all rows, not just the visible ones. |

Tests Document View

Tests > Document View

The **Document View** provides a high-level view of the important information, the status, and the last executions of all tests in the selected project.

For each test container, test folder, and test, the **Document View** displays the following columns:

| Item | Description |
|---------------------------------------|--|
| Last Execution | Time of the last execution of the selected test or project. |
| Status | <p>The status of the last test execution. If an execution is currently running, the status Not Executed is displayed.</p> <ul style="list-style-type: none">• Passed• Failed• Not Executed• N/A <p>For test containers and folders, a status bar displays the number of tests with the corresponding status.</p> |
| Last Build | Build on which the last execution was performed. |
| Last Finished Execution | Time of the last manual test execution that is finished. For automated tests, this status is the same as Last Execution . |
| Status Last Finished Execution | Status of the last manual test execution that is finished. For automated tests, this status is the same as Status . |
| Build Last Finished Execution | Build on which the last finished manual test execution was performed. For automated tests, this status is the same as Last Build . |
| Test Count | Number of tests. |
| Planned Time | Roll up of Planned Time of the manual tests. |
| Assigned Issues | Number of Issues assigned to the test. |
| Changed On | Last time the selected test element was changed. |
| Changed By | User that changed the selected test element for the last time. |
| Attributes | If you have assigned attributes to your tests, you can select to display them in the grid. Attributes are freely definable characteristics that can be applied to tests. You can use attributes for filtering and reporting purposes. |
| Workflow State | The workflow state assigned to the test. For more information, see <i>Test Asset Workflow</i> . |

Working With Tests in Grid View

The **Grid View** in the **Tests** area complements the **Document View** and **Details View** by facilitating the filtering, sorting, and grouping of large numbers of tests. **Grid View** eases identifying match points between tests and finding specific test information. Standard Windows keyboard shortcuts can be used, making it easy to select and manipulate specific sets of tests within **Grid View**. You can execute trial runs of tests and access tests from **Grid View**. You can also create execution plans through multi-selecting tests within **Grid View**.

Grid View offers a number of view-customization features that can help you better manage large numbers of tests. You can display or hide columns, adjust the width of columns, and move columns around using

drag-and-drop. To enhance performance when you have numerous tests, page views are broken into views of 50 tests each. You can advance through pages using the **First**, **Last**, **Next**, and **Previous**. Or you can enter a page number into the **Page** field and press **Enter** to access the specified page. Sorting, grouping, and filtering functions are available through context-menu commands to help you better organize your tests, group tests, and identify matching points between tests. All of your view-customization preferences are saved along with your project and will be available to you the next time you visit **Grid View**.

Accessing Tests from Grid View

You can access a test in **Tests** view directly from the **Grid View**.

To access a test's **Properties** page from **Grid View**:

1. In the menu, click **Tests > Grid View**.
2. Click on the **Name** of the test that you want to access.
Alternatively, right click the row of the test and select **Go to test**.

Creating an Execution Plan in Grid View

To create an execution plan in **Grid View**:

1. In the menu, click **Tests > Grid View**.
2. Select the tests you want to assign to your execution plan, by using the multi-select feature of the **Grid View**.
3. Right-click the tests and choose **Create Execution Plan**. The **New Execution Plan** dialog box displays.
4. Enter the specifications of your new execution plan.



Note: All selected tests must be in the same container. If not, the execution plan is not created and an error message displays. The test container is preselected in the **New Execution Plan** dialog box and can not be altered.

Modifying Properties and Attributes for Multiple Tests

To modify property or attribute values for multiple tests:

1. In the menu, click **Tests > Grid View**.
2. Select the tests for which you want to edit the properties or attributes. You can select multiple tests with **Ctrl+Click** or **Shift+Click**.
3. Right click and select **Multi-Edit**. The **Multi-Edit** dialog box opens. This dialog lets you set the value of the selected property or attribute to the same value for all selected tests.



Tip: Right-clicking on a cell of a specific property or attribute column pre-selects this property or attribute in the **Multi-Edit** dialog box.

4. Select **Properties** or **Attributes** from the **Category** list.
5. Select the **Name** of the property or attribute to modify.
6. Select the **Value** of the property or attribute that will apply to all selected tests.
7. Click **OK**. The selected property or attribute value is applied to all selected tests.



Tip: Multi-edit is also possible in the **Tests** tree (in the **Details View** and in the **Document View**).

Displaying and Hiding Columns in Grid View

To display and hide columns in **Grid View**:

1. In the menu, click **Tests > Grid View**.
2. Open a column header's context menu.

3. Expand the **Columns** submenu to view all the columns that are available in the project.
4. Check the check boxes of all the columns you want to have displayed in **Grid View**. Your column-display preferences will be saved and displayed each time you open the active project.

Grouping Tests in Grid View

Beyond simply sorting by column, you can chunk tests into groups to facilitate viewing. Groups are based on commonly-shared values within the column that the grouping is based on.

To group tests in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the column that the grouping is to be based on and select **Group by this field**. The tests are organized into groups based on commonly-shared values within the column you have selected.

Removing Grouping of Tests in Grid View

To remove the grouping of tests in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open any column header's context menu.
3. Uncheck the **Show in Groups** check box.

Sorting Tests in Grid View

To sort tests in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the column you want the tests to be sorted by.
3. Select **Sort Ascending** to have the tests sorted in ascending order or select **Sort Descending** to have the tests sorted in descending order. Your sort preferences are saved and displayed each time you open the active project.

Reordering Columns in Grid View

To reorder columns in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Select the column header of the column you want to move.
3. Drag the column to the desired position and release it. Your column-order preferences are saved and displayed each time you open the active project.

Resizing Columns in Grid View

To adjust the width of a column in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Select the vertical column-header divider of the column you want to adjust.
3. Drag the column boundary to the desired position and release it. Your column-width preferences are saved and displayed each time you open the active project.

Filtering Tests in Grid View

You can filter the test list based on column values. You can specify the following:

- Filter strings to be applied to text-based data fields.

- Calendar filters for date-based fields, using **Before**, **After**, or **On** operators.
- Numerical operators for number-based fields. **>**, **<**, and **=**.

Filtering Text-Based Values in Grid View

To filter text-based values in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the text-based column that the filter is to be based on.
3. Expand the **Filters** submenu on the context menu to display the **Filters** field.
4. Type the text string into the field.
5. Press **Enter**. All tests that match the filter criteria, for example, in the case of test names, all test names that include the specified string, are then dynamically displayed in the filtered list.

Filtering Date-Based Values in Grid View

To filter date-based values in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the date-based column that the filter is to be based on.
3. Hold your cursor over **Filters** on the context menu to display the **Before**, **After**, and **On** submenu.
4. Hold your cursor over **After** to define a date before which (and including) all tests should be excluded. Hold your cursor over **Before** to define a date after which (and including) all tests should be excluded. Hold your cursor over **On** to exclude all tests except those that have the specified date. The calendar tool displays.
5. Select a date using the calendar tool or click **Today** to specify today's date.



Tip: You must explicitly click a date on the calendar tool or click **Enter** to activate date-based filtering changes.

All tests that match the filter criteria are dynamically displayed in the filtered list.

Filtering Number-Based Values in Grid View

To filter number-based values in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the number-based column that the filter is to be based on.
3. Expand the **Filters** submenu on the context menu to display the **>**, **<**, and **=** operators.
4. Enter a number in the **>** field to define a number less than which (and including) all tests should be excluded. Enter a number in the **<** field to define a number greater than which (and including) all tests should be excluded. Enter a number in the **=** field to exclude all tests except those that have the specified number.



Note: Number values are rounded to two decimal places.

5. Press **Enter**. All tests that match the filter criteria are dynamically displayed in the filtered list.

Filtering Boolean Values in Grid View

To filter boolean values in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the boolean-based column that the filter is to be based on.
3. Expand the **Filters** submenu on the context menu to display the available values.
4. Click one of the **Yes** or **No** option buttons. All tests that match the filter criteria are dynamically displayed in the filtered list.

Filtering Values Using a Predefined List in Grid View

To filter values using a predefined list in **Grid View**:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the column that has a predefined filter value, for example `NodeType`, that the filter is to be based on.
3. Expand the **Filters** submenu on the context menu to display the available values.
4. Check the check boxes of the filter values that you are interested in. All tests having one of the selected criteria will be displayed.

Removing Specific Grid View Filters

 **Note:** You can identify filtered columns by their titles, which are displayed in bold, italic text. Hiding a column removes all filters that have been applied to the column.

To remove a specific **Grid View** filter:

1. In the menu, click **Tests > Grid View** .
2. Open the column header's context menu of the column that has the filter you want to remove.
3. Uncheck the **Filters** check box.

Removing All Grid View Filters

 **Note:** You can identify filtered columns by their titles, which are displayed in bold, italic text. Hiding a column removes all filters that have been applied to the column.

To remove all **Grid View** filters:

1. In the menu, click **Tests > Grid View** .
2. Open any column header's context menu and select **Reset Filters**.

Restoring Default Grid View Settings

Restoring default **Grid View** settings resets all user-defined settings for the current project. The user-defined settings are the following:

- Column order
- Column width
- Shown and hidden columns
- Applied filters
- Sorting
- Grouping

To restore the default **Grid View** settings:

1. In the menu, click **Tests > Grid View** .
2. Open any column header's context menu and select **Reset View**.

Assigning Existing Issues to Tests in Grid View

To assign existing issues to one or more tests:

1. In the menu, click **Tests > Grid View** .
2. Right-click on a test. You can select multiple tests with **Ctrl+Click** or **Shift+Click**.
3. Click **Assign Existing Issues**, select a **Profile** and type the issue numbers in the **Issue ID** field, separated by commas.

Execution Planning

The **Execution Planning** area enables you to maintain control over test executions during development and testing. This includes activities like configuring execution plans, scheduling execution plans, assigning tests to execution plans, setting up execution plan dependencies, configuring execution-server deployment, and configuring dynamic hardware-provisioning with keywords. Execution plans are displayed, organized, and maintained through a hierarchical tree structure, the **Execution Plans** tree. Each execution plan may have any number of child tests associated with it. The **Execution Plans** tree enables you to organize execution plans within folders, in any number of hierarchy levels.

Manual Execution Planning

To access this page, click **Execution Planning > Manual Execution Planning**.

A typical challenge of planning your manual testing is that there are too many tests and not enough resources. It is important to execute the tests with the most benefit. To support you here, the **Manual Execution Planning** page is divided into two views:

Test Selection Select manual tests and schedule them in testing cycles.

Test Assignment Plan the execution of manual tests in detail. Determine which manual tester has to execute which tests in a certain testing cycle.

To access the **Test Selection** view, click **Execution Planning > Manual Execution Planning**.

To access the **Test Assignment** view, click **Test Assignment** on the top right of the **Manual Execution Planning** page. To get back to the **Test Selection** view, click **Test Selection** on the top left.

Walkthrough

This walkthrough shows you features of **Manual Execution Planning** using an example. The assumptions for the example include:

- The requirements are fully documented and categorized according to a `Risk` property.
- The manual tests are fully documented, assigned to requirements, and categorized according to an `Importance` attribute which defines the necessity for executing a specific test.
- Quality goals, which are also called *project exit criteria*, have been defined based on the requirements property `Risk` and the test attribute `Importance`. These quality goals specify the levels of testing required.
- The only tests to plan are the ones that best fulfill the quality goals.

Overview (Walkthrough)

The **Manual Execution Planning** page consists of four areas presented on two screens. On the first screen, the **Test Selection** screen, you can select and schedule your tests for a testing cycle. On the second screen, the **Test Assignment** screen, you can assign the tests of the selected testing cycle to the testers who will execute them.

The **Test Selection** screen is divided into three areas:

- The **Filter Criteria** area on the left side. With the filter capabilities, you can narrow down the tests shown in the next area.
- The **Matching Tests** area in the middle. This list shows all available tests. When you apply filters, just the filtered tests are shown.

- The **Testing Cycles** area on the right side. Here you can assign tests from the list in the middle to a certain testing cycle, which is a defined time period.

To access the **Test Assignment** screen, click **Test Assignment >>>** on the top right. This screen is divided into two areas:

- The **Testing Cycles** area on the left side. This is the same area as on the **Test Selection** screen, it is shown on both screens.
- The **Assigned Tests** area on the right side. This list shows the tests, which have been added to the currently selected testing cycle. Here you can drag the tests from the list to the **Testing Cycles** area and drop them on a tester.

Creating a Testing Cycle (Walkthrough)

The first step is to create a testing cycle. A testing cycle is a defined period in time consisting of a start date, an end date, and a list of manual testers. We will name our testing cycle `Week 21 - Team A`.

1. In the menu, click **Execution Planning > Manual Execution Planning**.
2. In the **Testing Cycles** area on the right side, click **New Testing Cycle**. The **New Testing Cycle** box appears.
3. Enter a name for the testing cycle, in our case `Week 21 - Team A`.
4. Optionally enter a description.
5. Click  next to **Start date** to set the start date of the testing cycle.
6. Click  next to **End date** to set the end date of the testing cycle.
7. In `hh:mm` format, enter the amount of time that you want to provide for this testing cycle (**Capacity**), in our case, we entered `35:00`.
8. Click **OK**.

Our testing cycle now appears in the **Testing Cycles** area. It shows that currently no testers are assigned. The next step is to assign testers.

Adding Manual Testers (Walkthrough)

The next step is to add manual testers to our testing cycle. We are also going to define the capacity each manual tester has for this testing cycle.

1. In the header of our testing cycle `Week 21 - Team A`, click **... > Edit Assigned Testers**. The **Edit Assigned Testers** dialog box appears.
2. Click **Add/Remove Testers**. The **Add/Remove Testers** dialog box appears.
3. Select a user group from the **Available** list.
4. Select one or more users from the user group in the **Available** list.
5. Click **Add** or **Add All** to add users to the **Selected** list.
6. Click **OK**. The users are now added to the list in the **Edit Assigned Testers** dialog box.
7. For each user that was added, provide a capacity for the testing cycle in the **Capacity** field using the `hh:mm` format.
8. Click **Close**.

Now our testing cycle shows the available resources, both for the whole testing cycle as well as for each manual tester. A burn-down chart displays the assigned tests that are not yet completed. After we have created our testing cycle and added manual testers, we can determine which tests need to be done in the next step.

Selecting Tests and Scheduling (Walkthrough)

This topic describes how to assign tests to our testing cycle. The goal is to select those tests that best cover the defined quality goals. One of the quality goals is based on the requirement risk. The typical practice is to test things with high risk first.

1. In the **Filter Criteria** area on the left side, under **Quality Goals**, check the **Requirement Risk** check box. The list of **Matching Tests** is narrowed down to the tests that are critical.
2. In the **Matching Tests** area, open a column header's context menu to sort, filter or group tests. That helps to select the tests that should be scheduled.
3. Select one or more tests, drag them to the **Testing Cycle** area on the right side and drop them to our testing cycle. To assign tests to a testing cycle, you can also right-click on the selected tests and select **Assign tests to cycle**. When adding tests to a testing cycle, the indicator for already used resources is updated. Move the mouse over this indicator for detailed information about the overall capacity, the planned time, and the time left.
4. After scheduling all tests with a critical risk, schedule those with a high risk. After that, schedule those with a medium risk, and so on.

The tests assigned to our testing cycle have moved to the **No specific tester** row. In the next step, we will assign tests to manual testers.

Assigning Tests to Manual Testers (Walkthrough)

Now we are going to assign the tests in our testing cycle to manual testers.

1. Click **Test Assignment >>>** on the top right. The **Test Assignment** screen appears.
2. In the **Testing Cycles** area on the left side, click on the **No specific tester** row in our testing cycle. All tests that are not assigned to a tester appear in the **Assigned Tests** area on the right side.
3. Select one or more tests, drag them to the **Testing Cycles** area on the left side and drop them to a manual tester. When assigning tests to a manual tester, the resource indicator is updated. Move the mouse over this indicator to see detailed information about the capacity, the planned time, and the time left.
4. When you are finished assigning tests to manual testers, click ... > **Start Testing Cycle** to start the tests.

When the testing cycle is started, each manual tester is informed through email about the tests that are assigned. While the testing cycle is running, a tester will also receive an email when a new test is assigned to the tester or when a test from another tester is re-assigned to the tester. You can add a test inbox, (**Manual Tests Assigned to Me**), to your personal dashboard. All the tests that are assigned to you are listed there.

Test Selection

The **Test Selection** view is the initial view shown when you enter the **Manual Execution Planning** page. Here you can narrow down the list of manual tests with filters and quality goals, create and edit testing cycles, and assign the filtered manual tests to testing cycles.

The **Test Selection** view is divided into three areas:

- The **Filter Criteria** area on the left side. With the filters you can narrow down the tests shown in the next area.
- The **Matching Tests** area in the middle. This list shows all available tests. When you apply filters, just the filtered tests are shown.
- The **Testing Cycles** area on the right side. Here you can assign tests from the list in the middle to a testing cycle.

Filter Criteria

The **Filter Criteria** area offers the following filtering options:

Filter Summary Lists all filters applied to the page.

Click  next to a filter to remove the filter from the list.

Tests Filter Contains a list with all test filters of the current project.

Select a test filter. The filtered tests are shown in the **Matching Tests** area.

Quality Goals/ Assignment Progress Shows a list of the quality goals in the project and the values of the quality goals.

Select one or more quality goal values. The filtered tests are shown in the **Matching Tests** area.

Each quality goal value has an **Assignment Progress** bar. The bar shows the number of tests that are already assigned and the number of tests that are not yet assigned to an execution plan.



Attention: These numbers include manual and automated tests.

Move the mouse over a bar to show:

- The number of **Total matching tests**.
- The number of **Tests to reach goal**.
- The number of **Already assigned tests**.
- The number of **Not yet assigned tests**.

If **Already assigned tests** is greater than **Tests to reach goal** an exclamation point appears. This means that more tests are assigned than required.



Attention: These numbers include manual and automated tests.

Container/ Folder Displays the container/folder test hierarchy.

Select one or more containers and folders to display only the child tests of the selected nodes in the **Matching Tests** area. To select multiple containers and folders, use **Ctrl+Click** or **Shift+Click**.



Note: You can only select one single test filter and one goal value per quality goal.

If you select multiple quality goal values (from different quality goals) or a quality goal value and a test filter, only those tests that match all criteria are shown in the **Matching Tests** area.

Matching Tests

The **Matching Tests** area shows the manual tests that match the selected filter criteria. By default, no quality goals or filters are selected so all manual tests display in the grid.

Select filters and quality goal values in the **Filter Criteria** area to narrow down the list of manual tests.

The caption of the **Matching Tests** grid shows:

- The number of the currently selected tests.
- The total planned time of the currently selected tests.

You can perform the following actions in the grid:

- To assign a manual test to a testing cycle, drag the test onto the header of a testing cycle. You can also click the **Assign** action of a test or right-click the selected tests and then click **Assign Tests to Cycle**.

- To assign a manual test to a manual tester, drag the test onto the tester in the testing cycle.
- Open a column header's context menu to sort, to group and to filter manual tests, to add or remove columns and to reset filters or the view.
- Right-click on a manual test to show the test details or to print the manual test.
- You can select multiple tests with **Ctrl+Click** or **Shift+Click**.
- You can multi-assign tests to different testing cycles/testers. The column **Assigned** shows a green check mark if the test is assigned at least once.
- Click  to show the test details.
- Click  to show the test in the **Tests Details View**.

Testing Cycles

A testing cycle is a defined period in time consisting of a start date, an end date, and a list of manual testers. Create testing cycles to plan and execute specific sets of manual tests.

The **Testing Cycles** area lists all testing cycles of the current project. It is the central part of the **Manual Execution Planning** page and displays on both the **Test Selection** and on the **Test Assignment** view. The testing cycles are ordered chronologically.

Click **New Testing Cycle** on the top of the **Testing Cycles** area to create a new testing cycle. Click **Show/Hide Testing Cycles** if you want to hide testing cycles. Additionally, use the **Search** field to display only testing cycles that contain the search word. Hiding testing cycles helps you to keep the **Testing Cycles** area tidy and to enhance the performance. It is especially useful when you have lots of testing cycles.

In the header of a testing cycle, click **Actions (...)** to access the menu. Click  to locate the testing cycle in the **Execution Planning** tree.

In a testing cycle, click on a manual tester to display the assigned tests in the **Assigned Tests** area.

Click on the header of a testing cycle to select it. The tests that are assigned to the testing cycle (but not to a specific manual tester) are listed in the **Assigned Tests** area. If the **Assigned Tests** area is not displayed, click **Test Assignment** on the top right.

The testing cycle boxes show:

- The name of the testing cycle.
- The start and end dates.
- A burn-down chart that visualizes the testing progress. It shows how likely it is that all tests will be finished by the end of the testing cycle. Move your mouse over one of the bars to show the number of tests that are not completed.
- The capacity that is left. This value is calculated as follows: The sum of planned time of the assigned tests minus the capacity sum of the manual testers. This value displays both for the testing cycle and each individual tester. If a testing cycle or a manual tester is overloaded with tests, the green box turns into red. Move the mouse over this box to get detailed information.
- The number of assigned tests and the completed tests for both the testing cycle and each individual tester. Move the mouse over these numbers to get detailed information.
- The manual testers that are assigned to each testing cycle. Click on the number of the assigned testers to expand or collapse the detailed information.

Creating Testing Cycles



Note: You can also create new testing cycles in the **Execution Planning Details View** and **Document View**.

To create a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, click **New Testing Cycle**. The **New Testing Cycle** dialog box appears.

3. Enter a **Name** for the testing cycle.
4. Optionally enter a **Description**.
5. Click  next to **Start date** to set the start date of the testing cycle.
6. Click  next to **End date** to set the end date of the testing cycle.
7. In hh:mm format, enter the amount of time you want to provide for the testing cycle (**Capacity**).

This capacity will be replaced by the sum of the tester capacities as soon as you add a tester to the testing cycle.

8. Specify all **Required attributes**, if there are any. Required attributes must be specified when creating a new testing cycle. Required attributes that are not yet set display with an orange background and a red border.



Note: If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.



Note: If an attribute is set to **Inactive** (under **Execution Planning > Attributes**), it is not required when creating a new element.

9. Click **OK**. The testing cycle is added to the **Testing Cycles** area.

Adding Manual Testers to a Testing Cycle

After you have created a testing cycle, you may want to add manual testers and capacity to it.

You can also add manual testers in the **Execution Planning Details View**. In the menu, click **Execution Planning > Details View**. Click a testing cycle on the left side and click **Add/Remove Testers** on the **Properties** page.

To add manual testers to a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the header of a testing cycle in the **Testing Cycles** area, click ... > **Edit Assigned Testers**. The **Edit Assigned Testers** dialog box appears.
3. Click **Add/Remove Testers**. The **Add/Remove Testers** dialog box appears.
4. Select the user group you want to use from the **Available** list.
5. Select one or more users from the **Available** list.
You can select multiple users with **Ctrl+Click** or **Shift+Click**.
6. Click **Add** or **Add All** to add users to the **Selected** list.
7. Click **OK**. The users are added to the list of testers.
8. Define a capacity for each user of the testing cycle in the **Capacity** field using the hh:mm format.
9. Click **Close**. The testing cycle **Capacity** is updated with the cumulative value of the capacity for each manual tester.

Editing Assigned Testers

You can add/remove manual testers to/from testing cycles and define the capacity for each individual tester.

To edit assigned testers:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Edit Assigned Testers**. The **Edit Assigned Testers** dialog box appears.

3. Click **Add/Remove Testers**. The **Add/Remove Testers** dialog box appears.
4. Select the user group you want to use from the **Available** list.
5. Add or remove testers:
 - Select one or more users in the **Available** list and click **Add** or **Add All**.
 - Select one or more users in the **Selected** list and click **Remove** or **Remove All**.

You can select multiple users with **Ctrl+Click** or **Shift+Click**.
6. Click **OK**. The users are now added to or removed from the **Assigned Testers** list.
7. Define a capacity for each user of the testing cycle in the **Capacity** field using the `hh:mm` format.
8. Click **Close**. The testing cycle **Capacity** is updated with the cumulative value of the capacity for each manual tester.

You can also edit assigned testers in the **Details View**. In the menu, click **Execution Planning > Details View**. Click a testing cycle on the left side and click **Add/Remove Testers** on the **Properties** page.

Adding Milestones

A milestone is a target that must be reached by a specified date. Milestones are useful when the capacity of the testers is not spread evenly over the testing cycle. These milestones are represented by marks on the burn-up charts (on the dashboard) dispersed according to the date and capacity.

The following is an example of milestones within the context of a testing cycle:

- At least 20 percent of the tests must be completed by September 13th.
- At least 50 percent of the tests must be completed by September 15th.
- 100 percent of the tests must be completed by September 19th (end of the testing cycle).

To add a milestone:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click **... > Edit Milestones**. The **Edit Milestones** dialog box appears.
3. Click **Add Milestone**, define a due date and a percentage of completed tests, and click **OK**.
4. To edit a milestone, click . To delete a milestone, click .
5. Add additional milestones and click **Close**.



Note: If you duplicate a testing cycle or if you edit the start and end date, make sure the milestones are still within the defined period of the testing cycle.

Assigning Tests to Testing Cycles

To assign tests to a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Select tests in the **Matching Tests** area and drag them onto the header of a testing cycle. You can also click the **Assign** action of a test or right-click the selected tests and then click **Assign Tests to Cycle**.
You can select multiple tests with **Ctrl+Click** or **Shift+Click**.
The tests are added to the testing cycle.



Note: You can assign a test several times to different testing cycles/testers. The column **Assigned** in the **Matching Tests** area shows a green check mark if the test is assigned at least once.

Assigning Tests to Manual Testers



Note: Before you can assign tests to manual testers, you need to add manual testers to the testing cycle.

Assign tests to manual testers to distribute the workload among the testers.

You can assign a test several times to different testing cycles/testers. The column **Assigned** in the **Matching Tests** area shows a green check mark if the test is assigned at least once.

You can also assign tests to manual testers in the **Test Assignment** view. Click **Test Assignment** on the top right. Select tests in the **Assigned Tests** area and drag them onto a manual tester in a testing cycle.

To assign tests to manual testers:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In a testing cycle, click **Tester(s) assigned** to expand the list of manual testers.
3. Select tests in the **Matching Tests** area and drag them onto a manual tester in a testing cycle. You can also click the **Assign** action of a test or right-click the selected tests and then click **Assign Tests to Cycle**. Drop the selected tests onto the  icon to assign them to all testers.
You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

Moving Tests from One Testing Cycle to Another

You can move tests from one testing cycle to another testing cycle. That might be useful if you cannot finish all tests in time and you want to move those tests to the next testing cycle.

To move tests from one testing cycle to another testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.
3. In the **Testing Cycles** area on the left side, click on the header of a testing cycle.
4. Select tests in the **Assigned Tests** area and drag them onto another testing cycle. You can also click the **Reassign** action of a test or right-click the selected tests and then select **Reassign Tests**.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

Moving Tests from One Tester to Another

You can move tests from one manual tester to another with free capacity to distribute the workload within a testing cycle.

To move a test from one manual tester to another manual tester:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.
3. In the **Testing Cycles** area on the left side, click on a manual tester. The tests that are assigned to the manual tester are shown in the **Assigned Tests** area.
4. Select tests in the **Assigned Tests** area and drag them onto another manual tester. You can also click the **Reassign** action of a test or right-click the selected tests and then select **Reassign Tests**.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

Removing Tests from Testing Cycles

To remove tests from a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.
3. In the **Testing Cycles** area on the left side, click on the header of a testing cycle. The tests that are assigned to the testing cycle are shown in the **Assigned Tests** area.
4. Click  to remove a test from the testing cycle.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.



Note: You cannot remove tests from a finished testing cycle. You can copy a test from a finished testing cycle by dragging it onto another testing cycle or tester.

Removing Tests from Manual Testers

To remove tests from a manual tester:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.
3. In the **Testing Cycles** area on the left side, click on a manual tester. The tests that are assigned to the manual tester are shown in the **Assigned Tests** area.
4. You can remove tests from the testing cycle or remove them from just the manual tester but keep them in the testing cycle:
 - To remove a test from the manual tester and the testing cycle: click .
 - To remove a test from the manual tester but keep it in the testing cycle: click a test in the **Assigned Tests** area and drag it onto **No specific tester** in the **Testing Cycles** area.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.



Note: You cannot remove tests from a manual tester who is assigned to a finished testing cycle. You can copy a test from a finished testing cycle by dragging it onto another testing cycle or tester.

Importing Tests from Execution Plan

You need the **Manage testing cycles and configurations** permission to import tests from an execution plan.

To import tests from an execution plan:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click **... > Import Tests from Execution Plan**. The **Select Execution Plan** dialog box appears.
3. Select an execution plan and click **OK**. All tests of the execution plan are assigned to **No specific tester** of the testing cycle.

Synchronizing Testing Cycles

Synchronize testing cycles if you want to update the current runs of a testing cycle with the changes that were made to the assigned tests.

You need the **Manage testing cycles and configurations** permission to synchronize a testing cycle.

To synchronize a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Synchronize Testing Cycle**.
3. Confirm the message if you are sure that you want to update the current runs of the testing cycle with the changes that were made to the assigned tests.

Duplicating Testing Cycles

You need the **Manage testing cycles and configurations** permission to duplicate a testing cycle.

To duplicate a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Duplicate Testing Cycle**. The **Duplicate Testing Cycle** dialog box appears. The properties (**Name**, **Description**, **Start Date**, **End Date** and **Capacity**) are duplicated and filled in the fields.
3. *Optional:* Choose how you want to copy the test assignment. For example, you can choose not to copy **Passed** tests, enabling you to re-run the testing cycle only with **Failed** and **Not Executed** tests.
4. Click **OK**. The testing cycle is duplicated with all assigned tests and testers.

Hiding Testing Cycles

You can hide testing cycles on the **Manual Execution Planning** page to keep the page tidy and to enhance the performance. This is especially useful when you have lots of testing cycles.

To hide a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, click **Show/Hide Testing Cycles**. Additionally, use the **Search** field to display only testing cycles that contain the search word. The **Visible Testing Cycles** dialog box appears.
3. Uncheck all testing cycles that shall be hidden and click **OK**. To hide all finished testing cycles, click **Deselect Finished**.



Tip: Open a column header's context menu to sort, filter or group the testing cycles. When selecting multiple testing cycles, you can use the spacebar to toggle the selected items.

4. To quickly hide a certain testing cycle, you can also click ... in the header of a testing cycle and click **Hide Testing Cycle**.
5. To show a hidden testing cycle, again click **Show/Hide Testing Cycles** and mark the testing cycle.

If you hide a testing cycle on the **Manual Execution Planning** page, you can still see it in the **Execution Plans** tree in the **Details View** and in the **Document View** as well as in the dashboard panels and in the reports. The visibility settings are done per user. Every user can define which testing cycle to display and which to hide.

Starting Testing Cycles



Note: Before starting a testing cycle, verify that the start date of the testing cycle corresponds with the current date. If not, ensure that you are able to start the testing cycle on a different date.

To start a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.

2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Start Testing Cycle**.

A testing cycle can only be started once. If the testing cycle is already started, this action is disabled. A running testing cycle displays *Testing Progress* between the start and end date. A testing cycle that has not been started displays *Not Started*. A finished testing cycle displays *Finished*.

The **Start Testing Cycle** dialog box appears.

3. Select the **Version** and **Build** of the product that you want to test. All products that are specified in the assigned tests test container are listed.



Note: To change the version and build after a testing cycle has been started, click ... > **Update Versions and Builds**.

4. Click **OK**.

The assigned manual tests are distributed to the manual testers. The testers are notified by email and the tests will display in their **Manual Tests Assigned to Me** panel on their dashboard.

You can still change a running test, change the product version and build, assign additional tests, remove tests, or move tests between testers after a testing cycle has started.

Finishing Testing Cycles



Note: Before finishing a testing cycle, verify that the end date of the testing cycle corresponds with the current date. If not, ensure that you are able to finish the testing cycle on a different date.

To finish a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Finish Testing Cycle**.
If the testing cycle is not yet started, this action is disabled. A testing cycle that is not running displays *Not Yet Started* between the start and end date. A running testing cycle displays *Testing Progress*. A finished testing cycle displays *Finished*.
3. The **Finish Testing Cycle** dialog box appears. Select the action you want to perform for tests that have not been executed and click **Finish**. The selected status change will only be applied to non-executed tests for which no status reason has been set.



Note: Finished testing cycles are locked. You cannot add/remove tests or testers to/from a finished testing cycle. You can copy a test from a finished testing cycle by dragging it onto another testing cycle or tester.

Deleting Testing Cycles



Note: You can also delete testing cycles in the **Execution Planning Details View** and **Document View**. Right-click on a testing cycle and click **Delete**.

To delete a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the header of a testing cycle of the **Testing Cycles** area, click ... > **Delete Testing Cycle**.
3. Click **Yes** on the **Deletion Confirmation** dialog box.

Test Assignment

To access this page, click **Execution Planning > Manual Execution Planning**. On the top right, click **Test Assignment**.

The **Test Assignment** view is the second view of the **Manual Execution Planning** page. Here you can organize the manual tests that are assigned to the selected testing cycle.

Testing Cycles

A testing cycle is a defined period in time consisting of a start date, an end date, and a list of manual testers. Create testing cycles to plan and execute specific sets of manual tests.

The **Testing Cycles** area lists all testing cycles of the current project. It is the central part of the **Manual Execution Planning** page and displays on both the **Test Selection** and on the **Test Assignment** view. The testing cycles are ordered chronologically.

Click **New Testing Cycle** on the top of the **Testing Cycles** area to create a new testing cycle. Click **Show/Hide Testing Cycles** if you want to hide testing cycles. Additionally, use the **Search** field to display only testing cycles that contain the search word. Hiding testing cycles helps you to keep the **Testing Cycles** area tidy and to enhance the performance. It is especially useful when you have lots of testing cycles.

In the header of a testing cycle, click **Actions (...)** to access the menu. Click  to locate the testing cycle in the **Execution Planning** tree.

In a testing cycle, click on a manual tester to display the assigned tests in the **Assigned Tests** area.

Click on the header of a testing cycle to select it. The tests that are assigned to the testing cycle (but not to a specific manual tester) are listed in the **Assigned Tests** area. If the **Assigned Tests** area is not displayed, click **Test Assignment** on the top right.

The testing cycle boxes show:

- The name of the testing cycle.
- The start and end dates.
- A burn-down chart that visualizes the testing progress. It shows how likely it is that all tests will be finished by the end of the testing cycle. Move your mouse over one of the bars to show the number of tests that are not completed.
- The capacity that is left. This value is calculated as follows: The sum of planned time of the assigned tests minus the capacity sum of the manual testers. This value displays both for the testing cycle and each individual tester. If a testing cycle or a manual tester is overloaded with tests, the green box turns into red. Move the mouse over this box to get detailed information.
- The number of assigned tests and the completed tests for both the testing cycle and each individual tester. Move the mouse over these numbers to get detailed information.
- The manual testers that are assigned to each testing cycle. Click on the number of the assigned testers to expand or collapse the detailed information.

Creating Testing Cycles



Note: You can also create new testing cycles in the **Execution Planning Details View** and **Document View**.

To create a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, click **New Testing Cycle**. The **New Testing Cycle** dialog box appears.
3. Enter a **Name** for the testing cycle.
4. Optionally enter a **Description**.
5. Click  next to **Start date** to set the start date of the testing cycle.
6. Click  next to **End date** to set the end date of the testing cycle.
7. In hh:mm format, enter the amount of time you want to provide for the testing cycle (**Capacity**).

This capacity will be replaced by the sum of the tester capacities as soon as you add a tester to the testing cycle.

8. Specify all **Required attributes**, if there are any. Required attributes must be specified when creating a new testing cycle. Required attributes that are not yet set display with an orange background and a red border.



Note: If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.



Note: If an attribute is set to **Inactive** (under **Execution Planning > Attributes**), it is not required when creating a new element.

9. Click **OK**. The testing cycle is added to the **Testing Cycles** area.

Adding Manual Testers to a Testing Cycle

After you have created a testing cycle, you may want to add manual testers and capacity to it.

You can also add manual testers in the **Execution Planning Details View**. In the menu, click **Execution Planning > Details View**. Click a testing cycle on the left side and click **Add/Remove Testers** on the **Properties** page.

To add manual testers to a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the header of a testing cycle in the **Testing Cycles** area, click ... > **Edit Assigned Testers**. The **Edit Assigned Testers** dialog box appears.
3. Click **Add/Remove Testers**. The **Add/Remove Testers** dialog box appears.
4. Select the user group you want to use from the **Available** list.
5. Select one or more users from the **Available** list.
You can select multiple users with **Ctrl+Click** or **Shift+Click**.
6. Click **Add** or **Add All** to add users to the **Selected** list.
7. Click **OK**. The users are added to the list of testers.
8. Define a capacity for each user of the testing cycle in the **Capacity** field using the hh:mm format.
9. Click **Close**. The testing cycle **Capacity** is updated with the cumulative value of the capacity for each manual tester.

Editing Assigned Testers

You can add/remove manual testers to/from testing cycles and define the capacity for each individual tester.

To edit assigned testers:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Edit Assigned Testers**. The **Edit Assigned Testers** dialog box appears.
3. Click **Add/Remove Testers**. The **Add/Remove Testers** dialog box appears.
4. Select the user group you want to use from the **Available** list.
5. Add or remove testers:
 - Select one or more users in the **Available** list and click **Add** or **Add All**.

- Select one or more users in the **Selected** list and click **Remove** or **Remove All**.

You can select multiple users with **Ctrl+Click** or **Shift+Click**.

6. Click **OK**. The users are now added to or removed from the **Assigned Testers** list.
7. Define a capacity for each user of the testing cycle in the **Capacity** field using the `hh:mm` format.
8. Click **Close**. The testing cycle **Capacity** is updated with the cumulative value of the capacity for each manual tester.

You can also edit assigned testers in the **Details View**. In the menu, click **Execution Planning > Details View**. Click a testing cycle on the left side and click **Add/Remove Testers** on the **Properties** page.

Adding Milestones

A milestone is a target that must be reached by a specified date. Milestones are useful when the capacity of the testers is not spread evenly over the testing cycle. These milestones are represented by marks on the burn-up charts (on the dashboard) dispersed according to the date and capacity.

The following is an example of milestones within the context of a testing cycle:

- At least 20 percent of the tests must be completed by September 13th.
- At least 50 percent of the tests must be completed by September 15th.
- 100 percent of the tests must be completed by September 19th (end of the testing cycle).

To add a milestone:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click **... > Edit Milestones**. The **Edit Milestones** dialog box appears.
3. Click **Add Milestone**, define a due date and a percentage of completed tests, and click **OK**.
4. To edit a milestone, click . To delete a milestone, click .
5. Add additional milestones and click **Close**.



Note: If you duplicate a testing cycle or if you edit the start and end date, make sure the milestones are still within the defined period of the testing cycle.

Assigning Tests to Testing Cycles

To assign tests to a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Select tests in the **Matching Tests** area and drag them onto the header of a testing cycle. You can also click the **Assign** action of a test or right-click the selected tests and then click **Assign Tests to Cycle**.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

The tests are added to the testing cycle.



Note: You can assign a test several times to different testing cycles/testers. The column **Assigned** in the **Matching Tests** area shows a green check mark if the test is assigned at least once.

Assigning Tests to Manual Testers



Note: Before you can assign tests to manual testers, you need to add manual testers to the testing cycle.

Assign tests to manual testers to distribute the workload among the testers.

You can assign a test several times to different testing cycles/testers. The column **Assigned** in the **Matching Tests** area shows a green check mark if the test is assigned at least once.

You can also assign tests to manual testers in the **Test Assignment** view. Click **Test Assignment** on the top right. Select tests in the **Assigned Tests** area and drag them onto a manual tester in a testing cycle.

To assign tests to manual testers:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In a testing cycle, click **Tester(s) assigned** to expand the list of manual testers.
3. Select tests in the **Matching Tests** area and drag them onto a manual tester in a testing cycle. You can also click the **Assign** action of a test or right-click the selected tests and then click **Assign Tests to Cycle**. Drop the selected tests onto the  icon to assign them to all testers. You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

Moving Tests from One Testing Cycle to Another

You can move tests from one testing cycle to another testing cycle. That might be useful if you cannot finish all tests in time and you want to move those tests to the next testing cycle.

To move tests from one testing cycle to another testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.
3. In the **Testing Cycles** area on the left side, click on the header of a testing cycle.
4. Select tests in the **Assigned Tests** area and drag them onto another testing cycle. You can also click the **Reassign** action of a test or right-click the selected tests and then select **Reassign Tests**.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

Moving Tests from One Tester to Another

You can move tests from one manual tester to another with free capacity to distribute the workload within a testing cycle.

To move a test from one manual tester to another manual tester:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.
3. In the **Testing Cycles** area on the left side, click on a manual tester. The tests that are assigned to the manual tester are shown in the **Assigned Tests** area.
4. Select tests in the **Assigned Tests** area and drag them onto another manual tester. You can also click the **Reassign** action of a test or right-click the selected tests and then select **Reassign Tests**.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

Removing Tests from Testing Cycles

To remove tests from a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.

3. In the **Testing Cycles** area on the left side, click on the header of a testing cycle. The tests that are assigned to the testing cycle are shown in the **Assigned Tests** area.
4. Click  to remove a test from the testing cycle.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.



Note: You cannot remove tests from a finished testing cycle. You can copy a test from a finished testing cycle by dragging it onto another testing cycle or tester.

Removing Tests from Manual Testers

To remove tests from a manual tester:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.
3. In the **Testing Cycles** area on the left side, click on a manual tester. The tests that are assigned to the manual tester are shown in the **Assigned Tests** area.
4. You can remove tests from the testing cycle or remove them from just the manual tester but keep them in the testing cycle:
 - To remove a test from the manual tester and the testing cycle: click .
 - To remove a test from the manual tester but keep it in the testing cycle: click a test in the **Assigned Tests** area and drag it onto **No specific tester** in the **Testing Cycles** area.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.



Note: You cannot remove tests from a manual tester who is assigned to a finished testing cycle. You can copy a test from a finished testing cycle by dragging it onto another testing cycle or tester.

Importing Tests from Execution Plan

You need the **Manage testing cycles and configurations** permission to import tests from an execution plan.

To import tests from an execution plan:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click **... > Import Tests from Execution Plan**. The **Select Execution Plan** dialog box appears.
3. Select an execution plan and click **OK**. All tests of the execution plan are assigned to **No specific tester** of the testing cycle.

Synchronizing Testing Cycles

Synchronize testing cycles if you want to update the current runs of a testing cycle with the changes that were made to the assigned tests.

You need the **Manage testing cycles and configurations** permission to synchronize a testing cycle.

To synchronize a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click **... > Synchronize Testing Cycle**.
3. Confirm the message if you are sure that you want to update the current runs of the testing cycle with the changes that were made to the assigned tests.

Duplicating Testing Cycles

You need the **Manage testing cycles and configurations** permission to duplicate a testing cycle.

To duplicate a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Duplicate Testing Cycle**. The **Duplicate Testing Cycle** dialog box appears. The properties (**Name**, **Description**, **Start Date**, **End Date** and **Capacity**) are duplicated and filled in the fields.
3. *Optional:* Choose how you want to copy the test assignment. For example, you can choose not to copy **Passed** tests, enabling you to re-run the testing cycle only with **Failed** and **Not Executed** tests.
4. Click **OK**. The testing cycle is duplicated with all assigned tests and testers.

Hiding Testing Cycles

You can hide testing cycles on the **Manual Execution Planning** page to keep the page tidy and to enhance the performance. This is especially useful when you have lots of testing cycles.

To hide a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
 2. In the **Testing Cycles** area, click **Show/Hide Testing Cycles**. Additionally, use the **Search** field to display only testing cycles that contain the search word. The **Visible Testing Cycles** dialog box appears.
 3. Uncheck all testing cycles that shall be hidden and click **OK**. To hide all finished testing cycles, click **Deselect Finished**.
-  **Tip:** Open a column header's context menu to sort, filter or group the testing cycles. When selecting multiple testing cycles, you can use the spacebar to toggle the selected items.
4. To quickly hide a certain testing cycle, you can also click ... in the header of a testing cycle and click **Hide Testing Cycle**.
 5. To show a hidden testing cycle, again click **Show/Hide Testing Cycles** and mark the testing cycle.

If you hide a testing cycle on the **Manual Execution Planning** page, you can still see it in the **Execution Plans** tree in the **Details View** and in the **Document View** as well as in the dashboard panels and in the reports. The visibility settings are done per user. Every user can define which testing cycle to display and which to hide.

Starting Testing Cycles



Note: Before starting a testing cycle, verify that the start date of the testing cycle corresponds with the current date. If not, ensure that you are able to start the testing cycle on a different date.

To start a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.
2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Start Testing Cycle**.

A testing cycle can only be started once. If the testing cycle is already started, this action is disabled. A running testing cycle displays **Testing Progress** between the start and end date. A testing cycle that has not been started displays **Not Started**. A finished testing cycle displays **Finished**.

The **Start Testing Cycle** dialog box appears.

3. Select the **Version** and **Build** of the product that you want to test. All products that are specified in the assigned tests test container are listed.



Note: To change the version and build after a testing cycle has been started, click ... > **Update Versions and Builds**.

4. Click **OK**.

The assigned manual tests are distributed to the manual testers. The testers are notified by email and the tests will display in their **Manual Tests Assigned to Me** panel on their dashboard.

You can still change a running test, change the product version and build, assign additional tests, remove tests, or move tests between testers after a testing cycle has started.

Finishing Testing Cycles



Note: Before finishing a testing cycle, verify that the end date of the testing cycle corresponds with the current date. If not, ensure that you are able to finish the testing cycle on a different date.

To finish a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.

2. In the **Testing Cycles** area, in the header of a testing cycle, click ... > **Finish Testing Cycle**.

If the testing cycle is not yet started, this action is disabled. A testing cycle that is not running displays *Not Yet Started* between the start and end date. A running testing cycle displays *Testing Progress*. A finished testing cycle displays *Finished*.

3. The **Finish Testing Cycle** dialog box appears. Select the action you want to perform for tests that have not been executed and click **Finish**. The selected status change will only be applied to non-executed tests for which no status reason has been set.



Note: Finished testing cycles are locked. You cannot add/remove tests or testers to/from a finished testing cycle. You can copy a test from a finished testing cycle by dragging it onto another testing cycle or tester.

Deleting Testing Cycles



Note: You can also delete testing cycles in the **Execution Planning Details View** and **Document View**. Right-click on a testing cycle and click **Delete**.

To delete a testing cycle:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.

2. In the header of a testing cycle of the **Testing Cycles** area, click ... > **Delete Testing Cycle**.

3. Click **Yes** on the **Deletion Confirmation** dialog box.

Manual Configuration Testing

You can execute a manual test with different software and hardware configurations.

To assign configurations to a manual test:

1. In the menu, click **Execution Planning > Manual Execution Planning**. The **Manual Execution Planning** page displays.

2. Click **Test Assignment** on the top right. The **Manual Execution Planning** page switches into the **Test Assignment** view.

3. Right-click a test in the **Assigned Tests** area.

You can select multiple tests with **Ctrl+Click** or **Shift+Click**.

4. Click **Specify Configurations**. The **Specify Configurations** dialog box appears.

5. Click > or < to assign or unassign the selected configurations. You can also create a new configuration by entering a new name in the **Select or enter configurations** field and clicking >.

To edit and organize your configurations in greater detail, click **Execution Planning > Configurations**. For each configuration, an instance of the test is added to the **Assigned Tests** grid.



Note: If you unassign a configuration, this configuration will be disabled for the currently selected tester. However, instances assigned to other testers will not be removed.

6. Click **OK**.

You can assign the new test instances to different manual testers and testing cycles.

Manual Testing

The manual testing functionality of Silk Central allows testers to efficiently execute manual tests. The **Manual Testing** window provides an intuitive, easy-to-use UI with just the information that is relevant to the manual tester in the execution process.

Manual testers have quick access to the **Manual Testing** window via the dashboard panel **Manual Tests Assigned to Me**. For more information on the dashboard and on panels, see *Dashboard*. The **Manual Testing** window is structured as follows: In the overview area, it lists the tests that are contained by the selected execution plan or testing cycle. Beneath, in the test steps area, it displays the list of steps that are configured for the selected test. Manual testers can process the steps from the top to the bottom.

As a manual tester, you can tick off tests and test steps when you have executed them successfully, write a result text, create and assign issues, and attach result files. Additionally, you can capture screen images and record videos to illustrate how you executed a test or test step, or to show an issue that occurred during the testing process.

The integrated timer shows you, how long it took you to execute a test and a bar visualizes the testing progress. The **Testbook** (which is a panel you can add to your dashboard) shows you real-time updates about all activities that take place during manual testing within a project.

Manual Testing Window

To open the **Manual Testing** window, click **Home > My Dashboard** in the menu, and then click  (**Continue Manual Test**) in the panel **Manual Tests Assigned to Me**. If this panel does not display on your dashboard, you have to add it. For more information on the dashboard and on panels, see *Dashboard*.

You can also access the **Manual Testing** window on the **Activities** page (in the menu, click **Tracking > Activities**) in the **Current Executions** grid, or on the **Current Run** page.

The **Manual Testing** window shows two areas:

- The overview area on the top.
- The test steps area on the bottom.

Overview Area

The overview area is the area on the top of the **Manual Testing** window.

The overview area shows the following information:

| Item | Button | Description |
|---|---|---|
| Execution plan or testing cycle | | The name of the execution plan or testing cycle that contains the tests. |
| Edit Settings > Selection Behavior on Status Change |  | For each individual user, you can configure if the next test/step should automatically be selected when the status of a test/step is changed, |

| Item | Button | Description |
|---|---|---|
| <p>Edit Settings > Visibility of Test Information</p> |  | <p>click Edit Settings and click one of the following in the Edit Settings dialog:</p> <ul style="list-style-type: none"> • Always select next test/step • Select next test/step on passed only • Keep current test/step selected <p>For each individual user, you can configure which additional test information is displayed in the overview area, click Edit Settings and check the corresponding check boxes in the Edit Settings dialog. The following additional information can be displayed:</p> <ul style="list-style-type: none"> • Test ID • Description.  Note: For business process tests, checking this check box will additionally display the description for each business component in the business process test. • Status and Status Reason  Note: For business process tests, checking this check box will additionally display the status for each business component in the business process test. • Configuration • Project name • Path info • Keywords • Executed by • Mobile Device Selection • Run comment • Test Attributes • Execution Plan Attributes |
| <p>Edit Settings > Visibility of Test Actions</p> |  | <p>For each individual user, you can hide or display the following actions.</p> <ul style="list-style-type: none"> • Edit version and build • Start/stop code analysis • Synchronize test run |
| <p>Edit Settings > Video Recording Options</p> <p>Finish</p> |  | <p>For each individual user, you can check or uncheck the Include Audio to define if videos should be recorded with or without audio.</p> <p>Click Finish to finish the run. This option displays only if you open a test that is unassigned or that is assigned to an execution plan. It does not display if you open a test that is assigned to a testing cycle. Select the action you want to perform for tests that have not been executed and click Finish. The selected status change will only be applied to non-executed tests for which no status reason has been set.</p> |
| <p>Passed</p> |  | <p>Click the button to mark the status of the currently selected test as Passed and automatically select the next test. This is the default setting of the button.</p> |
| <p>Failed</p> |  | <p>Click the button to mark the status of the currently selected test as Failed and automatically select the next test.</p> |

| Item | Button | Description |
|-------------------------------|---|--|
| Test name | | The list that appears shows all tests of the execution plan or testing cycle and their current status. Click on a test or press the Up and Down keys to navigate through the tests and their steps. If you specified to run a test with a certain configuration, the configuration name displays next to the test name in parenthesis. For more information, see <i>Manual Configuration Testing</i> . Click  to show the test in the Details View of the Tests area. |
| Requirements Assigned |  | Click the button to view the requirements that were assigned to the test. The button displays only if there is at least one requirement assigned. The number beside the button shows how many requirements were assigned. |
| Issues Assigned |  | Click the button to view the issues that were assigned to the test. The button displays only if there is at least one issue assigned. The number beside the button shows how many issues were assigned. The Issues dialog box shows both the issues that were assigned to the test and the issues that were assigned to the test steps. |
| Attachments Assigned |  | Click the button to view the attachments that were assigned to the test. The button displays only if there is at least one attachment assigned. The number beside the button shows how many attachments were assigned. |
| Block/Unblock |  | Click the button to set the test to blocked/unblocked. Setting a test to blocked is useful to indicate that the test cannot be continued due to certain restrictions, for example a problem with the IT infrastructure. |
| Test Status |  | The status of the test. It can be set automatically or manually. Click  beside the status icon to select a status. For more information, see <i>Statuses of Tests and Test Steps</i> . |
| Edit Version and Build |  | Click the button to edit the version and the build. For more information, see <i>Editing Version and Build</i> . |
| Run Comment | | Enter a high-level comment for the test run. This helps to speed up the test execution review process, as the comment is displayed in the test run grids. |
| Capture Screen |  | Click the button to capture an image of your screen and to attach it to the test. The captured image will be added to the attachment list. For more information, see Capturing Screen Images . |
| Record Video |  | Click the button to record a video of the actions on your screen and to attach it to the test. The video will be added to the attachment list and you can save it to your local file system. For more information, see Recording Videos . |
| Attach Result File |  | Click the button to attach a result file to the test. For more information, see Attaching Result Files . |
| Assign or Create Issue |  | Click the button to create a new issue and attach it to the test or to assign an existing issue. For more information, see Assigning or Creating Issues . |
| Select Device |  | Select a mobile device (physical device, emulator or simulator) on which to execute your manual test. For more information, see Selecting Mobile Devices . |

| Item | Button | Description |
|-----------------------------|---|---|
| Start Code Analysis |  | Click the button to start the analyzing process. The Start Code Analysis dialog box displays. Select a Profile and enter one or more Hostnames . If coverage information already exists for this test, you can replace it by checking the checkbox. Click Start Code Analysis to actually start the process. When you click Stop Code Analysis , Silk Central stores the code analysis information to the test run. |
| Synchronize Test Run |  | Click the button to synchronize a test run with an adapted test. If you started to execute a test and you decide to make changes to this test in the Tests unit in the meantime, you can synchronize the test run with the adapted test.  Note: This button is not visible for business components in business process tests as business components cannot be synchronized. |

For business process testing, the overview area provides the following additional information:

| Item | Button | Description |
|---------------------------|--------|---|
| Business Component | | The currently active business component. In contrast to manual testing, where all test steps can be executed at the same time, only one component in a business process test is executed at a time. |

 **Note:** To close a dialog box, click anywhere outside or click  (**Close**).

Test Steps Area

The test steps area is below the overview area on the **Manual Testing** window.

The test step area shows:

| Item | Button | Description |
|---------------------------------|---|---|
| Edit Used Time |  | Click the button to manually enter how long it took you to execute the test. When you open the Manual Testing window or if you select another test, the timer automatically starts. The flashing colon indicates that the timer is running. Click  to pause the timer, click  to start the timer. The figures on the left side show the actually used time, the figures on the right side show the planned time. |
| Progress Bar | | Visualizes the testing progress. The progress bar shows how many steps you have already executed. |
| Post Message to Testbook |  | Click the button to post a message to the Testbook . |
| Passed |  | Click the button to mark the status of the currently selected test step as Passed and automatically select the next step. This is the default setting of the button. |
| Failed |  | Click the button to mark the status of the currently selected test step as Failed and automatically select the next step. This is the default setting of the button. |
| Show/Hide Details |  | Click the button to expand a test step and to display detailed information like the Action Description or the Expected Results as well as all buttons and the result field. |
| Step name | | The name of the test step. Step names with the status Not Executed are bold. |

| Item | Button | Description |
|-------------------------------|--|--|
| Issues Assigned |  | Click the button to view the issues that are assigned to the test step. The button displays only if there is at least one issue assigned. The number beside the button shows how many issues are assigned. |
| Attachments Assigned |  | Click the button to view the attachments that are assigned to the test step. The button displays only if there is at least one attachment assigned. The number beside the button shows how many attachments are assigned. |
| Step Status |  | The status of the step. Click ▼ beside the status icon to select a status. Silk Central will automatically calculate a test status based on the step statuses. For more information, see <i>Statuses of Tests and Test Steps</i> . |
| Result field | | Enter result text into the field. Click anywhere outside to save the text. |
| Capture Screen |  | Click the button to capture an image of your screen and to attach it to the test step. The captured image will be added to the attachment list. For more information, see Capturing Screen Images . |
| Record Video |  | Click the button to record a video of the actions on your screen and to attach it to the test step. The video will be added to the attachment list and you can save it to your local file system. For more information, see Recording Videos . |
| Attach Result File |  | Click the button to attach a result file to the test step. For more information, see Attaching Result Files . |
| Assign or Create Issue |  | Click the button to create a new issue and attach it to the test step or to assign an existing issue. For more information, see Assigning or Creating Issues . |



Note: To close a dialog box, click anywhere outside or click  (Close).



Tip: You can use the keyboard when you work on a test: To change the status of the selected test step, use the shortcuts **Ctrl+Shift+A** (for passed) and **Ctrl+Shift+S** (for failed). These combinations can easily be pressed with one hand. Press the **Up** and **Down** keys to navigate through the tests and their steps, press **Right** to expand a step and **Left** to collapse it. To close a dialog box, press **Esc**.

Mobile Device User Interface

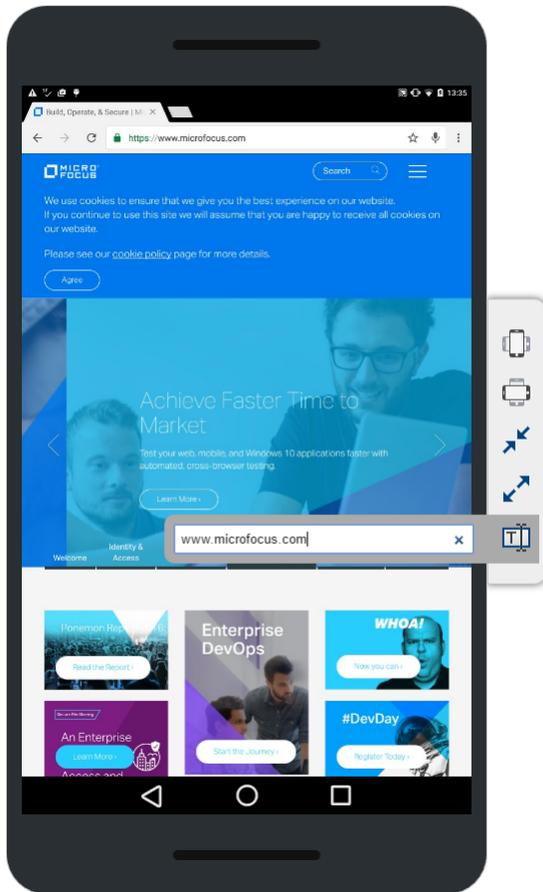
When you select a mobile device to execute your manual test, you are presented with the mobile device user interface.

Using your mouse, you can interact with the emulated display of the device. Use the icons on the right-hand side of the display to rotate the device or to simulate the pinch-in and pinch-out actions.

Use the text icon to enter text into the currently focused field. The text icon also supports the input method editor (IME) for entering multibyte characters.

iOS Limitations:

- You can only run one simulator per user session, but multiple physical devices can execute tests simultaneously.
- The *Home* button is not supported when testing on Sauce Labs devices.
- If a device becomes unavailable, close and restart the device stream to continue your test.



Capturing Screen Images

When you execute tests in the **Manual Testing** window, you can capture images of your screen. The captured images will be attached to the test or test step. This is especially helpful when a test step fails and you want to outline when, where, and how the issue occurred.

Make sure that you have a Java Runtime Environment (JRE) installed and that the application is allowed to run. You can download JRE from [Java SE Downloads](#). If your browser does not support NPAPI, Silk Central automatically offers to download and run a JNLP file, which is required to enable this feature. If you cannot download the JNLP file (for example due to security restrictions), refer to [Manually Setting Up JNLP Files](#) on page 255.

To capture a screen image:

1. Open the **Manual Testing** window. For more information, see *Manual Testing Window*.
2. Click  (**Show Details**) on the left side to expand a test step.
- 3.



Click  (**Capture Screen**). If your browser does not support NPAPI, Silk Central automatically offers to download and run a JNLP file, which is required to enable this feature. If you cannot download the JNLP file (for example due to security restrictions), refer to [Manually Setting Up JNLP Files](#) on page

255. Once the JNLP application runs, click  again. The **Screen Capturing** functionality starts.

4. Perform one of the following actions:

- Draw a rectangle with your mouse to select the area that shall be captured.
- Press **Enter** to take a capture of the entire screen. If you have more than one monitor, the screen of all monitors will be captured.
- Press **Esc** to cancel.

The **Screen Capturing** window appears.

5. Edit the screen capture as desired. You can add text and graphical items like arrows and rectangles. In the editor, you can use the keyboard to perform the following actions:
 - Press the arrow keys to move objects (like rectangles or arrows) around. Hold down **Shift** and press the arrow keys to precisely adjust the position.
 - Select an object with the **Move** tool and press **Backspace** or **Delete** to remove the object.
 - If you use the text tool, you can press **Shift+Enter** to confirm the input or press **Esc** to cancel.
 - Press **Ctrl+C** to copy the screen capture to the clipboard.
6. Click **Upload**. The screen capture is attached to the test step.

7.

To view the attached screen capture, click . The **Attachments** dialog box with a list of all attached files appears.

8. You can delete attached files by clicking  (**Delete Result File**) or open the files by clicking  (**Open**).



Note: In this dialog box, you can only delete files you have attached during the current test run. You cannot delete files that were assigned to the test before the testing cycle/execution plan was started.



Note: You can perform the task described in this topic both on test level and on step level.

Recording Videos

When you execute tests in the **Manual Testing** window, you can record videos of the testing process. This is especially helpful when a test step fails and you want to outline when, where, and how the issue occurred.

The video will be attached to the test step where you started the recording from. For example: If you start the recording from step two and you continue with step three and four while the recording is yet running, the video will still be attached to step two.

Make sure that you have a Java Runtime Environment (JRE) installed and that the application is allowed to run. You can download JRE from [Java SE Downloads](#). If your browser does not support NPAPI, Silk Central automatically offers to download and run a JNLP file, which is required to enable this feature. If you cannot download the JNLP file (for example due to security restrictions), refer to [Manually Setting Up JNLP Files](#) on page 255.

Limitations:

- Video recording is only supported on Microsoft Windows client machines.
- Video capturing on Microsoft Windows 10 N and KN editions is not supported.
- Video recording of WPF windows is not supported on Windows 7 Classic Theme.
- Recording videos at 4k resolution is not supported.

To record a video:

1. Open the **Manual Testing** window. For more information, see *Manual Testing Window*.
2. Click  (**Show Details**) on the left side to expand a test step.

3.



Click  (**Record Video**). If your browser does not support NPAPI, Silk Central automatically offers to download and run a JNLP file, which is required to enable this feature. If you cannot download the

JNLP file (for example due to security restrictions), refer to [Manually Setting Up JNLP Files](#) on page

255. Once the JNLP application runs, click  again.

If suitable audio devices are detected, the video will be recorded with audio. If multiple devices are connected, the first one detected will be used. Click  to enable the audio setting, as it is turned off by default. You need to re-start the video recording after changing this setting.

The **Video Recording** window appears and the recording automatically starts.

4. When you are done, click **Stop Video Recording**.

5. Enter a **Result filename** and click **Upload File** to attach the video to the test step. You can also click **Save File** to save it to your local computer.

6. Click **Close**.

7.

To view the attached video, click . The **Attachments** dialog box with a list of all attached files appears.

8. You can delete attached files by clicking  (**Delete Result File**) or open the files by clicking  (**Open**).

 **Note:** In this dialog box, you can only delete files you have attached during the current test run. You cannot delete files that were assigned to the test before the testing cycle/execution plan was started.

 **Note:** You can perform the task described in this topic both on test level and on step level.

Attaching Result Files

When you execute tests in the **Manual Testing** window, you can attach result files to each test step.

To attach a result file:

1. Open the **Manual Testing** window. For more information, see *Manual Testing Window*.

2. Click  (**Show Details**) on the left side to expand a test step.

3. Click  (**Attach Result File**). The system file manager opens.

4. Browse for the file that you want to attach and confirm. The file is attached to the test step.

5.

To view the attached file, click . The **Attachments** dialog box with a list of all attached files appears.

6. You can delete attached files by clicking  (**Delete Result File**) or open the files by clicking  (**Open**).

 **Note:** In this dialog box, you can only delete files you have attached during the current test run. You cannot delete files that were assigned to the test before the testing cycle/execution plan was started.

 **Note:** You can limit the size of the result files that testers are allowed to upload. For more information, see *Setting the Maximum Size of Result Files*.

 **Note:** You can perform the task described in this topic both on test level and on step level.

Assigning or Creating Issues

When you execute tests in the **Manual Testing** window, you can assign issues to each test step.

To assign an issue:

1. Open the **Manual Testing** window. For more information, see *Manual Testing Window*.
2. Click  (**Show Details**) on the left side to expand a test step.
3. Click  (**Assign or Create Issue**). The **Assign or Create Issue** dialog box appears.
4. If you want to assign existing issues, click **Assign existing issues** and type the issue numbers in the **Issue ID** field, separated by commas.
5. If you want to create a new issue, click **Create new issue**. Enter information about the issue in the fields and select values from the lists.
6. Click **OK**. The issue is assigned to the test step.
7. To view the assigned issue, click . The **Issues** dialog box with a list of all attached issues appears.
8. If you click  (**Open Issue**), you are directed to an external issue tracking tool or to Issue Manager. To delete an issue, click  (**Delete Issue**).

 **Note:** In this dialog box, you can only delete issues you have assigned during the current test run. You cannot delete issues that were assigned to the test before the testing cycle/execution plan was started.

 **Note:** When you create a new issue, Silk Central automatically creates links to all result files that are currently assigned to the step or to the test. These links display in the description of the issue in your issue tracking tool. This way, you can directly download result files out of your issue tracking tool.

 **Note:** You can perform the task described in this topic both on test level and on step level.

Selecting Mobile Devices for Manual Testing

Before you can select a mobile device (physical device, emulator or simulator) on which to execute your manual test, a device provider needs to be configured. Note that the device provider and your current project must be assigned to the same location, otherwise the devices will not show up in the list. For additional information, see [Managing Device Providers](#).

 **Tip:** If you want to open the browser on the selected device with a specific URL, define a parameter called `silkTest.url` and enter your URL in the **Value** field.

To select a mobile device:

1. Open the **Manual Testing** window. For more information, see *Manual Testing Window*.
2. Click  (**Select Device**). The **Select Device** dialog box appears.
3. Select a mobile device. Use the **Search** field, the **Platform** and the **Application** lists on the left-hand side to filter the list of available devices. To install a native app on the mobile device for testing, select **Native App** in the **Application** list on the left-hand side and specify a URL or a UNC path to the app in the field below (iOS requires a URL). Example: `\\fileserver\apps\testapp.apk`

 **Tip:** Use the search field to search for specific device types, for example by searching for `Emulator or Simulator`.
4. *Optional:* Check the **Use physical devices only** check box if you want to exclude emulators and simulators for your device selection.
5. *Optional:* Uncheck the **Hide used or blocked devices** check box to see who is currently using devices.
6. Click **Start Device** to begin executing your test on the selected device.

Starting Code Analysis

When you execute tests in the **Manual Testing** window, you can start code analysis to deliver information about what code is covered by the manual test. For information on how to enable code analysis, see [Enabling Code Analysis](#).

Make sure that you have a Java Runtime Environment (JRE) installed and that the application is allowed to run. You can download JRE from [Java SE Downloads](#). If your browser does not support NPAPI, Silk Central automatically offers to download and run a JNLP file, which is required to enable this feature. If you cannot download the JNLP file (for example due to security restrictions), refer to [Manually Setting Up JNLP Files](#) on page 255.

To start code analysis:

1. Open the **Manual Testing** window. For more information, see *Manual Testing Window*.
2. Click  (**Show Details**) on the left side to expand a test step.

3.



Click **(Start Code Analysis)**. If your browser does not support NPAPI, Silk Central automatically offers to download and run a JNLP file, which is required to enable this feature. If you cannot download the JNLP file (for example due to security restrictions), refer to [Manually Setting Up JNLP Files](#) on page 255. The **Start Code Analysis** dialog box appears.

4. Select a **Profile** and enter one or more **Hostnames**. If coverage information already exists for this test, you can replace it by checking the checkbox. Click **Start Code Analysis** to actually start the process.
5. When you are done, click **Stop Code Analysis**. The code analysis information is automatically attached to the test run.

6.

To view the attached code analysis information, click . To view the code analysis report, select **Reports > Code Analysis**.



Note: You can perform the task described in this topic both on test level and on step level.

Editing Version and Build

To edit the version and the build in the **Manual Testing** window, click  (**Edit Version and Build**) in the overview area. If the button does not display, click  (**Edit Settings**) and enable the action.

The tests of a testing cycle might not be done with only one build, but with several ones. This is especially probable if you have long testing cycles with many tests. For example: The tests 1 and 2 of a testing cycle are executed with build 315, test 3 of the same testing cycle is executed with build 316.

The configured build and version are stored for each test and display in all relevant grids within Silk Central.

When you set a build for a test, this build is automatically set for all tests ...

- that are assigned to you or that are not assigned to any specific tester, and
- that have not yet been executed (tests that have the status `Not Executed`), and
- that belong to the same project and product.

So you do not have to set the same build again and again for each test that you are executing subsequently.

Statuses of Tests and Test Steps

When you set the status of a test step in the **Manual Testing** window, the status of the test changes. The following table shows how Silk Central calculates the test status depending on the statuses of the test steps:

| Status | Description |
|---|---|
|  Passed | If all steps are passed, the test status is set to <code>Passed</code> . No status reasons from test steps are propagated to a test's status. |

| Status | Description |
|---|---|
|  Failed | If all steps are executed and at least one step is failed, the test status is set to <code>Failed</code> . The status reason of the first failed test step is used as the failed status reason for the test. |
|  Not Executed | As long as all steps are not executed, the test status is set to <code>Not Executed</code> . The status reason of the first not-executed test step is used as the not executed status reason for the test. If you change the status of at least one step, the test status is set to <code>In Progress</code> . |
|  In Progress | Until every step is executed, the test status is set to <code>In Progress</code> . |

You can also set the test status manually. In the overview area of the **Manual Testing** window, or in the test steps area of a test step, click  (**Select Status**) beside the status icon and select a status. See *Status Reasons* for detailed information about defining reasons for each status.

 Apart from the test's status, you can also set a test to being blocked. Setting a test to blocked is useful to indicate that the test cannot be continued due to certain restrictions, for example a problem with the IT infrastructure.

Note the following:

- A manually set test status overwrites the automatically calculated one.
- If you change a step status afterwards, the test status is calculated again.
- You can manually set a test status back to `Not Executed`. However, Silk Central will never automatically set back the test status to `Not Executed`.
- You can manually set a test status to `In Progress`, even if you do not change any step status.

Printing Manual Tests

You can print a form of each test and fill in the result, the used time, the status, and other information by hand.

To print a manual test:

1. In the menu, click **Home > My Dashboard**.
2. In the panel **Manual Tests Assigned to Me**, click  (**View Manual Test Results**) in the **Actions** column. The **Manual Test Results View** appears.
3. Click **Print** or **Download as PDF**.

Testbook

The **Testbook** is a dashboard panel that gives you real-time updates about all activities that take place during manual testing. It shows you who did what and when in the different testing cycles and execution plans. The **Testbook** simplifies the coordination and collaboration among testers and test managers.

You can filter the entries by user, action, and testing cycle. The panel shows the activities of just one project. To configure a project for the panel, click  (**Configure**) in the header of the panel. To view activities in other projects, you can add more panels. For more information on panels, see *Dashboard*.

For each entry the user name is displayed and (if available) the full name. For organizations that do not use meaningful user names, the full name of a user is better recognizable for the other users within a team.

On the **Manual Testing** window, you can post a message to the **Testbook** by clicking .

Entries older than 60 days are deleted automatically. To change this default setting, open the file `Conf \AppServer\TMApserverHomeConf.xml` and edit the following setting:

```
<MaximumActivityLogLifeTimeInDays>60</MaximumActivityLogLifeTimeInDays>
```

The following activities are logged in the **Testbook**:

| Icon | Activity | Icon | Activity |
|---|------------------------|---|-----------------|
|  | Testing cycle started |  | Test added |
|  | Testing cycle finished |  | Test started |
|  | Testing cycle removed |  | Test finished |
|  | Milestone added |  | Test reassigned |
|  | Milestone edited |  | Test removed |
|  | Milestone removed |  | Test blocked |
|  | Mobile device selected |  | Message posted |

Manually Setting Up JNLP Files

Downloading and installing JNLP and JAR files can be a problem for users, for example due to security restrictions. Also, not all JDKs contain JNLP.

An alternative is to download the JAR file and to deploy it to client machines through the command line or a batch file. You can download the JAR file from: `http://<server>:19120[/instance_name]/ManualTesting/capture/manualtestingapplet.jar`

To launch the application from the command line, use the following command:

```
java -cp manualtestingapplet.jar
com.borland.sctm.manualtesting.ManualTestingApplet 19150 "Manual Testing |
Silk Central" "Keep this application running in the background to enable
screen and video capturing." "en"
```

This way, the manual testing application can be automatically started or testers can start it on demand using a desktop shortcut.

Offline Manual Testing

Offline Manual Testing allows you to download tests and complete them offline. It also allows you to upload your results back to Silk Central.

Testers use the **Manual Tests Assigned to Me** panel to download manual tests and upload results back to Silk Central. Test Managers use the **Offline Manual Testing Overview** panel to view and manage downloaded tests.

 **Note:** A **Manual Testing** license is checked out when a test is downloaded for offline execution. The license is checked back in when results are uploaded.

Downloading Manual Tests for Offline Execution

This topic will show you how to download tests and enter the **Status**, **Reason** and **Results** while you are offline.

1. Go to **Home > My Dashboard**. Ensure that the **Manual Tests Assigned to Me** panel is visible.
2. Select one or more tests. Right click your selection and choose **Download Selected Manual Tests for Offline Execution**. You can also select a single test and choose **Download all Manual Tests of the Related Execution Plan for Offline Execution**.



Tip: Make use of the **Rows per page** setting and the sorting and filtering capabilities to receive the list of tests that you need to select.

3. In the **Download Manual Tests for Offline Execution** dialog box, specify how you want the tests to be organized in Excel:
 - Create a separate worksheet for each test with a list of steps
 - Create a single worksheet with a list of tests without steps
4. If your tests contain attributes, select **Include Attributes**. All attributes are included in the downloaded Excel sheet, enabling you to filter your tests by each attribute.
5. Select **Include Attachments** to also download files that are attached to the selected tests and test steps. Note that uploading attachments back to Silk Central is not supported.
6. Click **Download**. Download and save the Excel file.
7. Open the file and, depending in which format you downloaded the Excel file, enter the **Status** and the **Result** or **Used Time** for each step or for each test.
8. Save the file.

Uploading Offline Manual Test Results

Once your downloaded manual tests are complete, use the following procedure to upload the results to Silk Central.

1. Go to **Home > My Dashboard**. Ensure that the **Manual Tests Assigned to Me** panel is visible.
2. Click **Upload Offline Manual Test Results**. The **Upload Offline Manual Test Results** dialog box displays.
3. Click **Choose File** and browse to your Excel file. The **Open** dialog box displays.
4. Select the file to upload and click **Open**.
5. Click **Upload**. A dialog box will display indicating the status of the upload.
6. Click **OK**.

Configuration Testing

Configuration testing is the process of testing a system with each of the supported software and hardware configurations.

The **Execution** area supports configuration testing by allowing reuse of the assigned tests. You can create configuration suites with a set of assigned tests, and all execution plans that you add to the configuration suite will also have the set of tests assigned. You can also create configuration suites from existing execution plans and copy and paste or cut and paste execution plans in the **Execution** tree into a configuration suite. Silk Central enables you to add or remove parameters, keywords, and manual testers to or from the configurations. When you create a configuration suite out of an existing execution plan, all the results of the execution plan are preserved in the configuration suite. If you copy and paste an execution plan into an existing configuration suite, these results are not preserved.

Each execution plan in the configuration suite is displayed in an editable grid. You can view the execution plans or configurations that contain a specific test in the **Properties** tab of the test. You can also view the

execution plans or configurations that are associated with a specific requirement in the **Assigned Tests** tab of the requirement.

To define configurations for automated tests, use the **Configurations Suite Configurations** page. To define configurations for manual tests use the **Configurations** page.

Creating a Configuration Suite

To create a configuration suite:

1. In the menu, click **Execution Planning > Details View** or **Execution Planning > Document View**.
2. Right-click the root node or a folder in the **Execution** tree and select **New Child Configuration Suite** or click  in the toolbar. The **New Configuration Suite** dialog box displays.
3. Type a **Name** and **Description** for the new configuration suite.
4. Select a **Test Container** from the list box.
5. Select a **Version** and **Build** from the list boxes, or check the **Read from Build Information File** check box to read the version and build from a file.
6. *Optional:* In the **Source Control Label** field you can specify that earlier versions of automation files, instead of the latest versions, are fetched from the source control system.



Note: The **Source Control Label** property is only enabled if the associated test container uses a source control profile that supports versioning.

7. Specify all **Required attributes**, if there are any. Required attributes must be specified when creating a new configuration suite. Required attributes that are not yet set display with an orange background and a red border.



Note: If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.



Note: If an attribute is set to **Inactive** (under **Execution Planning > Attributes**), it is not required when creating a new element.

8. Click **OK**.

Creating a Configuration Suite from an Execution Plan

To create a configuration suite from an existing execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, right-click the execution plan from which you want to create a configuration suite, and select **Convert to Configuration Suite**. Silk Central creates a new configuration-suite node, with the same name as the selected execution plan, in the **Execution** tree and adds the execution plan as a configuration sub-node to the configuration suite.

When you create a configuration suite out of an existing execution plan, all the results of the execution plan are preserved in the configuration suite. If you copy-paste an execution plan into an existing configuration-suite, these results are not preserved.

Adding Parameters to a Configuration

To add a predefined parameter to a configuration:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, select the configuration node to which you want to add a parameter.

3. Click the **Configurations** tab.
4. Click **Add/Remove Parameters**. The **Add/Remove Parameters** dialog box opens. The **Assigned parameters** list box on the right side of the dialog box lists all of the parameters that are defined for the configuration.
5. Select parameters in the **Select or enter parameters** list box that reflect your execution environment requirements.
Use **Ctrl+Click** or **Shift+Click** to select multiple parameters using standard browser multi-select functions.
6. To add any of the parameters listed in the **Select or enter parameters** list box, select the parameter and click **>**.
7. To create a new parameter, type the name of the parameter in the **Enter parameters** field, and press **Enter** or click **>**.
 **Tip:** The **Enter Parameters** field is auto-complete enabled. When you enter alphanumeric characters into this field, the field is dynamically updated with an existing parameter that matches the entered characters. The field is disabled when multiple parameters are selected in the **Select parameters** or **Assigned parameters** list boxes.
8. Click **OK**. The **Add/Remove Parameters** dialog box closes and a new column is added to the grid for each new parameter.

Removing Parameters from a Configuration

To remove a predefined parameter from a configuration:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, select the configuration node from which you want to remove a parameter.
3. Click the **Configurations** tab.
4. Click **Add/Remove Parameters**. The **Add/Remove Parameters** dialog box opens. The list box on the right side of the dialog box lists all of the parameters that are defined for the configuration.
5. To remove a parameter from the list box, select the parameter and click **<**.
6. Click **OK**. The **Add/Remove Parameters** dialog box closes and the column of the removed parameter is removed from the grid.

Assigning Keywords to a Configuration

To assign keywords to a configuration:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the configuration suite that you want to edit.
3. Click the **Configurations** tab.
4. Click into the **Keywords** column of the configuration that you want to add a keyword to. The **Assign Keywords** dialog box opens.
5. Select keywords in the **Select keywords** list box that reflect your execution environment requirements.
Use **Ctrl+Click** or **Shift+Click** to select multiple keywords using standard browser multi-select functions.

 **Tip:** The **Enter keywords** field is auto-complete enabled. When you enter alphanumeric characters into this field, the field is dynamically updated with an existing keyword that matches the entered characters. The field is disabled when multiple keywords are selected in the **Select keywords** or **Assigned keywords** list boxes.

 **Tip:** For automated execution plans, if you only have a few execution servers and do not require hardware provisioning, it might be enough to use only the default, reserved keywords that are

created for each execution server. In such cases, it is not necessary that you select additional keywords.

6. Click **>** to move the keyword into the **Assigned keywords** list box.



Note: For automated execution plans, the execution servers that match the assigned keywords are listed below in the dynamically-updated **Matching execution servers** list box. This list updates each time you add or remove a keyword. Click on the name of an execution server in the list to access the execution servers in **Administration > Execution Servers**.

7. Click **OK** to save the keywords and close the **Assign Keywords** dialog box.

Removing Keywords from a Configuration

To remove keyword assignments from a configuration:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the configuration suite that you want to edit.
3. Click the **Configurations** tab.
4. Click into the **Keywords** column of the configuration that you want to remove a keyword from. The **Assign Keywords** dialog box opens.
5. Select the no longer needed keywords in the **Assigned keywords** list.
Use **Ctrl+Click** or **Shift+Click** to select multiple keywords.
6. Click **<** to remove the keyword assignments.
7. Click **OK** to save the keywords and close the **Assign Keywords** dialog box.



Note: Keywords that are not in use anymore are automatically deleted from the system.

Adding Manual Testers to Configurations

For configurations that include manual tests, the **Configurations** page enables you to assign users as manual testers to a selected configuration. You can assign multiple manual testers to the configuration.

To assign a manual tester to a configuration:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the configuration suite that you want to edit.
3. Click the **Configurations** tab.
4. Click into the **Manual Testers** column of the configuration that you want to add a manual tester to. The **Manual Testers** dialog box opens.
5. In the **Available** list box, select the user group of which the tester is a member. The list box is populated with all members of the user group.
6. Select the name of the user you want to assign as a manual tester.
7. Click **Add** to add the user to the **Selected** list box; or click **Add All** to add all of the group's members and testers.
8. Click **OK** to close the **Add/Remove Testers** dialog box.

Removing a Tester Assignment from a Configuration

To remove a tester assignment from a configuration:

1. In the menu, click **Execution Planning > Details View**.
2. Select the configuration node from which you want to remove a tester assignment.
3. Click the **Configurations** tab.

4. Click into the **Manual Testers** column of the configuration that you want to remove a manual tester from. The **Manual Testers** dialog box opens.
5. From the **Selected** list, select the name of the assigned user that you want to remove.
6. Click **Remove** to remove the user from the **Selected** list; or click **Remove All** to remove all tester assignments.
7. Click **OK** to close the **Add/Remove Testers** dialog box.

Assigning Mobile Devices to a Configuration

To assign mobile devices to a configuration:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the configuration suite that you want to edit.
3. Click the **Configurations** tab.
4. Select a mobile device, device filter, or a native app:
 - To select a specific device, use the list on the right-hand side.
 - To select a device filter, select **Any device** in the list on the right-hand side. Edit this filter using the **Search** field, the **Platform** and the **Application** lists on the left-hand side.
 **Tip:** Use the search field to search for specific device types, for example by searching for `Emulator` or `Simulator`.
 - To install a native app on the mobile device for testing, select **Native App** in the **Application** list on the left-hand side and specify a URL or a UNC path to the app in the field below (iOS requires a URL). Example: `\\fileserver\apps\testapp.apk`
5. *Optional:* Check the **Use physical devices only** check box if you want to exclude emulators and simulators for your device selection.
6. *Optional:* Check the **Hide used or blocked devices** check box to only list devices that are currently available for use. This is especially useful if you want to execute a manual test, or if you want to run an execution plan immediately.
7. Click **OK** to save your mobile device selection.

During execution, either the selected device or the first matching device of a filter will be used and blocked until the execution plan has finished. If the selected device is not available or a filter does not match any available devices, the execution job stays in the queue until a suitable execution server and mobile device becomes available. Sauce Labs devices are not actually single devices, but rather device types, so they are not blocked during execution.

Configurations Suite Configurations Page

To access this page, click **Execution Planning > Details View > <Configuration Suite> > Configurations**.

The **Configurations** tab displays the configurations that are included in a configuration suite. The grid provides data for the automated generation of child execution plans.

Each configuration that you add to the grid results in a new execution plan, which is executed with the configuration suite.

To define a new configuration, click **New Configuration**. To add or remove a parameter, click **Add/Remove Parameters**.

For each listed configuration, the page displays the following columns:

| Column | Description |
|--------------------------------|---|
| Actions | <ul style="list-style-type: none"> Click  to delete the configuration. Click  to duplicate the configuration. |
| Name | The name of the configuration. Click on the name to change it. |
| Keywords | The list of the execution keywords that are assigned to the configuration. Click on the link to change the list. For additional information, see Assigning Keywords to a Configuration . |
| Manual Testers | The list of the manual testers that are assigned to the configuration. Click on the link to change the list. For additional information, see Adding Manual Testers to Configurations . |
| Mobile Device Selection | The mobile device (physical device, emulator or simulator) that is assigned to the configuration. Click the link to edit the mobile device selection. For additional information, see Assigning Mobile Devices to a Configuration . |
| Parameters | The grid can have multiple parameter columns. Click Add/Remove Parameters to add or remove parameters from the grid. To add a value to a parameter, click the field and type in the value. For additional information, see Adding Parameters to a Configuration . |

Configurations Page

To access this page, click **Execution Planning > Configurations**.

The **Configurations** page shows a list of the defined configurations in a grid. You can use these configurations for manual tests only. To define configurations for automated tests, see [Configurations Suite Configurations Page](#).

To define a new configuration, click **New Configuration**. To add or remove a parameter, click **Add/Remove Parameters**.

For each listed configuration, the page displays the following columns:

| Column | Description |
|--------------------------------|---|
| Actions | <ul style="list-style-type: none"> Click  to delete the configuration. Click  to duplicate the configuration. |
| Name | The name of the configuration. Click on the name to change it. |
| Mobile Device Selection | The mobile device (physical device, emulator or simulator) that is assigned to the configuration. Click the link to edit the mobile device selection. For additional information, see Assigning Mobile Devices to a Configuration . |
| Keywords | The list of the execution keywords that are assigned to the configuration. Click on the link to change the list. For additional information, see Assigning Keywords to a Configuration . |
| Parameters | The grid can have multiple parameter columns. Click Add/Remove Parameters to add or remove parameters |

| Column | Description |
|--------|---|
| | from the grid. To add a value to a parameter, click the field and type in the value. For additional information, see Adding Parameters to a Configuration . |

Managing Execution Plans, Folders, and Configuration Suites

An execution plan is a collection of assigned tests that are stored in a single test container. An execution plan can be run at configurable schedules and deployed on specified execution servers. Use folders to structure your execution plans, configuration suites, and testing cycles. You can also nest folders within folders.

The process of adding and editing execution plans is the same for both automated execution plans and manual execution plans.

Creating Execution Plans

To create an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. Select a folder or the project node in the **Execution Plans** tree.
3. Click  (**New Child Execution Plan**) on the toolbar or right-click the folder, testing cycle, or node and click **New Child Execution Plan**. The **New Execution Plan** dialog box appears.
4. Type a name and description for the execution plan.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. Select a test container from the **Test Container** list.



Note: You can assign only one test container to an execution plan. Only tests from this test container can be executed.

The latest version and build that are defined within the product that the container belongs to are populated automatically in the **Version** and **Build** lists.

6. Select a product **Version** and **Build** from the lists.

These are used when a new run of this execution plan is started. Alternatively, you can check the **Read from Build Information File** check box. In this case the version and build are read from the build information file on the application server at the beginning of each run. If a build information file is available on the execution server, this file is used by default for the test run, overriding the settings on the **New Execution Plan** dialog box.

7. Select a **Priority** for the execution plan from the list.

For automated execution plans, when more than one execution plan is queued but only one execution server is available, the **Priority** determines which execution plan is executed first. The same is true if execution servers are available but multiple execution plans require an unavailable mobile device.

8. In the **Source Control Label** field you can optionally specify that the execution plan is of an earlier version than the latest version. The label must refer to a version in the source control system that the test container is associated with. If this field is left blank, the latest version of the execution plan will be fetched. The **Source Control Label** property is only enabled if the associated test container uses a source control profile that supports versioning. Make sure to have enough free disk space on the execution server or servers when working with multiple versions of source files. Each version will be saved in its own folder on every execution server.



Tip: If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan.

- Specify all **Required attributes**, if there are any. Required attributes must be specified when creating a new execution plan. Required attributes that are not yet set display with an orange background and a red border.



Note: If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.



Note: If an attribute is set to **Inactive** (under **Execution Planning > Attributes**), it is not required when creating a new element.

- Click **OK** to update the **Execution** tree with the newly created execution plan.

Creating an Execution Plan in Grid View

To create an execution plan in **Grid View**:

- In the menu, click **Tests > Grid View**.
- Select the tests you want to assign to your execution plan, by using the multi-select feature of the **Grid View**.
- Right-click the tests and choose **Create Execution Plan**. The **New Execution Plan** dialog box displays.
- Enter the specifications of your new execution plan.



Note: All selected tests must be in the same container. If not, the execution plan is not created and an error message displays. The test container is preselected in the **New Execution Plan** dialog box and can not be altered.

Test Status Calculation

Each execution plan has one of the following status conditions:

| Status | Condition |
|---------------------|---|
| Passed | All considered test execution plans have the status Passed . |
| Failed | At least one considered test execution plan has the status Failed , but none of the execution plans has the status Not Executed . |
| Not Executed | At least one considered test has the status Not Executed . |

A test gets its status from the result of the latest execution plan run. If you manually change the status of the latest execution plan run, the test status changes also.



Note: If the latest execution plan run is deleted, the status of the test resets to the status of the latest existing execution plan run. If the deleted execution plan run was the only existing execution plan run, the status of the test is set to **N/A**, as if the test was newly created.

Execution Plan Attributes

Attributes are freely definable characteristics that can be applied to execution plans. You can use attributes for filtering and reporting purposes.



Note: Inheritance of attributes is similar to inheritance of properties and success conditions. Attributes that are assigned to a parent node are inherited throughout all sub-folders and child execution plans. Attributes cannot be defined for the project node and the **Unassigned Tests** node.

Assigning Attributes to Execution Plans

To assign an attribute to an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan to which you want to assign an attribute.
3. In the **Attributes** section, click **Add Attribute**. The **Add Attributes** dialog box appears.
4. Click **+** in the **Add** column of the attribute that you want to assign. Based on the attribute type you have selected, an **Edit Attribute** dialog box appears, where you can specify which of the available attribute values you want to assign to the execution plan.
5. Select an attribute value and click **OK** to assign the attribute to the execution plan.

Editing Execution Plan Attributes

To edit an execution plan attribute:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan for which you want to edit an assigned attribute.
3. In the **Attributes** section, click the **Edit** icon of the attribute you want to edit. The **Edit Attribute** dialog box displays. The available options in the **Edit Attribute** dialog box vary depending on the attribute type that you have selected.
4. Select the required value and click **OK** to save your settings.

Deleting Attributes from Execution Plans

To delete an attribute from an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan from which you want to delete an assigned attribute.
3. In the **Attributes** section, click the **Delete** icon of the attribute you want to delete. The **Delete Attribute** confirmation dialog box displays.
4. Click **Yes** to delete the attribute.



Note: Inherited attributes and attributes that were already used in a test run cannot be deleted.

Execution Plan Parameters

To apply a specific configuration to an execution plan, you can override the parameters that you have assigned to a test in the **Parameters** section on the **Properties** page of the execution plan. For additional information on test parameters, see *Test Parameters*.

Predefined Parameters

Silk Central provides information on execution plan run properties during test executions as parameters.

These parameters can be read by your test automation tool as environment variables (see [Test Parameters](#)). Additionally, any JUnit test class can access a test parameter of the underlying test as a Java system property (see [Accessing Silk Central Parameters through Java System Properties](#)). Use the `AttributeGet` methods to read these parameters in a Silk Performer script.

These parameter values can also be referenced in placeholders, as described in [Parameter-Token Replacement Upon Execution](#).

The following predefined parameters can be accessed:

| Parameter | Description |
|---|---|
| #sctm_execdef_name | The name of the execution plan that is currently executed. If the execution was triggered from the Tests area, the parameter has the value <code>Unassigned Tests</code> . |
| #sctm_execdef_id | The numeric identifier (database key) of the execution plan that is currently executed. |
| #sctm_product | The name of the product as defined in the executed test container. |
| #sctm_version | The name of the version to which the results of the execution are associated. |
| #sctm_build | The name of the version to which the results of the execution are associated. |
| #sctm_keywords | A comma separated list which contains the keywords that are defined for this execution plan. |
| #sctm_regular_execdef_run_id | The numeric identifier (database key) of the regular execution plan run. For setup and cleanup tests this property contains the run ID of the regular run. |
| #sctm_test_results_dir | The path of the directory where the result files of the test reside. |
| #sctm_test_name | The name of the Silk Central test being executed. |
| #sctm_test_id | The numeric identifier (database key) of the test being executed. |
| #sctm_source_root_dir | The local path of the test container's root directory, where all files under version-control reside. |
| #external_id | The unique identifier of a test case in a test package to be executed. You can use this as a filter if only specific test cases in a test package should be executed. |
| #is_test_package_root | If <code>true</code> , the test being executed is the root node of the test package (default). If <code>false</code> , only specific test cases in a test package specified by <code>#external_id</code> are executed |
| #sctm_data_driven_parent_test_name | The name of the parent test, if the executed test is a data driven row. |
| #sctm_data_driven_parent_test_id | The numeric identifier (database key) of the parent test, if the executed test is a data driven row. |

Creating an Execution Plan Parameter

To add a new parameter to an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, select the execution plan to which you want to add a new parameter.
3. In the **Parameters** section, click **New Execution Plan Parameter**. The **New Execution Plan Parameter** dialog box displays.
4. Type the **Name** and **Value** for the new parameter into the corresponding fields. New execution plan parameters are always of the type string and you cannot define a different type.
5. Click **OK** to add the parameter to the execution plan.

 **Note:** You cannot create execution plan parameters with the same name as an existing test parameter, which is included in the execution plan.

For configuration testing, you can create multiple execution plans that include the same tests and you can then define different configurations in the **Parameters** page. Parameters from different tests that have the same name and type are displayed only once in the page. If multiple parameters from different tests have the same name, you can only override one of these parameters.

Overriding a Test Parameter

If multiple parameters of different types but with the same name exist in tests, you can override only one of them. Only the values of parameters with the same type and name are replaced with the overriding value during execution.

To override a test parameter in an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, select the execution plan in which you want to override a test parameter.
3. In the **Parameters** section, click  in the **Actions** column of the parameter that you want to override. The **Set Execution Plan Parameter Value** dialog box displays.
4. Type the new **Value** into the field.
5. Click **OK**.

 **Note:** Click  in the **Actions** column of the parameter to undo the override.

Removing an Execution Plan Parameter

 **Note:** You cannot remove test parameters in an execution plan. You can only remove execution plan parameters.

To remove an execution plan parameter:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, select the execution plan from which you want to remove a parameter.
3. In the **Parameters** section, click  in the **Actions** column of the parameter you want to remove.

Setup and Cleanup Tests

The pre-test setup and post-test cleanup functionality of Silk Central enables you to define a setup test and a cleanup test for each execution plan. Setup tests are typically built upon scripts or manual procedures that prepare testing environments in anticipation of tests. Cleanup tests typically include scripts or manual procedures that restore test environments to their original state following tests. When setup tests are aborted, regular tests do not execute, however cleanup tests do execute to restore the testing environment to its original state.

You must create your setup and cleanup tests before you can assign them to execution plans. Any test can serve as a setup or cleanup test, except parent tests of multiple data-driven test instances. They require no special configuration and can be either automated or manual. The only requirement is that they perform the required setup and cleanup processes within your test environment. In the case of automated tests these are scripts that perform required setup and cleanup tasks. In the case of manual tests, these are manual setup and cleanup tasks.

 **Note:** The challenge in executing setup and cleanup tests is preventing their results from being aggregated with the results of the regular tests that they support. Silk Central addresses this concern by running setup and cleanup tests, both automated and manual, in independent execution plans,

thereby isolating actual test results from incidental performance fluctuations that may be caused by setup and cleanup tests.

Combining Automated and Manual Tests

Silk Central supports execution plans that include combinations of automated tests and manual tests. For such execution plans, Silk Central withholds execution of regular tests, both automated and manual, until setup tests are complete. Silk Central also ensures that all regular tests are complete before cleanup tests are run.

When manual tests are combined with automated tests, automated tests, on all execution servers, do not begin until the setup processes are complete. In the case of manual setup tests, regular automated tests begin only after manual setup routines are complete.

Configuring Setup and Cleanup Executions

To define a test as a setup or cleanup test:

1. In the menu, click **Execution Planning > Details View**.
2. Click the execution plan for which you want to configure a setup or cleanup test.
3. In the **Test Execution Setup** section, choose between a setup or cleanup test:
 - To define a setup test, click  (**Edit**) in **Setup Test**. The **Edit Setup Test** dialog box appears.
 - To define a cleanup test, click  (**Edit**) in **Cleanup Test**. The **Edit Cleanup Test** dialog box appears.

You cannot assign a test to the same execution plan as both a setup test and a regular or cleanup test. Assigned tests can come from any test container within your project. It is therefore possible to assign tests that have associated products and source control profiles that vary from their host execution plans.

4. Select a test in the **Tests** tree.
5. Click **OK**.

When failed tests are rerun, the corresponding setup/cleanup routines are also rerun. Setup/cleanup tests are not run with **Trial Run** test runs because such executions do not rely on execution plans.

Configuring Deployment Environments

Execution Deployment Page

Execution Planning > Details View > <Execution Plan> > Deployment

The **Deployment** page displays all of the hardware-provisioning keywords that are defined for this execution plan. These keywords are used to describe the execution environment requirements for the execution plan. An execution server only matches the selected automated execution plan if it has all keywords assigned the execution plan requires. The **Deployment** page also displays the users who are assigned to execute manual tests, as well as the Silk Test Classic AUT hosts that are assigned to execute Silk Test Classic tests.



Note: New execution servers are set-up in **Administration > Execution Servers**. For details on how to set-up an execution server, see [Setting Up Execution Servers](#).

Manual Test Execution

Lists all manual testers who are assigned to this execution plan or folder. Click **Edit** to edit the list of manual testers. For additional information, see [Adding or Removing Manual Testers](#).

Automated Test Execution

The **Deployment** page displays the following items in the **Automated Test Execution** section:

| Item | Description | | | | | | | | | | |
|--|---|-------|-------------|-----|---|--------------|--|---------------|---|-----------------|--|
| Assigned keywords | <p>Lists the keywords that are assigned to the execution plan.</p> <p>Automated execution plans Keywords are used to automatically identify an appropriate execution server for each test execution.</p> <p>External execution Assigned keywords are only considered to select an execution server for automated setup and cleanup tests.</p> <p>Manual execution plans Keywords are used by the manual tester to reflect the test environment.</p> <p>For additional information, see Assigning Keywords to Execution Plans.</p> | | | | | | | | | | |
| Keyword-related list of potential execution servers for deployment at execution time | <p>Lists the active execution servers that have keyword lists that match the keywords list of this execution plan. All keywords in the keywords list of the execution plan must be included in the keyword list of the execution server. Click on the name of an execution server in the list to access the execution server list in Administration > Execution Servers.</p> | | | | | | | | | | |
| Exclusive execution of assigned tests | <p>Exclusive execution defines whether assigned tests of other execution plans having this setting disabled can be executed on the same matching execution server at the same time (Disabled), or if only tests of one single execution plan can run at the same time (Enabled). Disabling exclusive execution reduces the required execution time, and is especially useful for mobile testing, where execution plans can execute on multiple devices on a device provider at the same time. No changes to your current execution configuration are required when disabling exclusive execution. For additional information, see Exclusive Execution.</p> | | | | | | | | | | |
| Distributed execution of assigned tests | <p>Distributed execution defines whether tests of an execution plan are executed sequentially on the selected execution server (Disabled), or if they are distributed to all matching execution servers and run in parallel (Enabled). No changes to your current execution configuration are required when turning on distributed execution. For additional information, see Distributed Execution.</p> | | | | | | | | | | |
| External execution of assigned tests | <p>Switching on external execution mode for an execution plan enables you to execute test suites in an external third- party system while maintaining the corresponding automated tests and results in Silk Central. For additional information, see External Execution.</p> | | | | | | | | | | |
| Video Recording | <p>Defines whether a video is recorded during the execution of the execution plan. This setting overrides the video recording settings of the individual tests that are assigned to the execution plan. You can select one of the following values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>---</td> <td>Video recording is disabled, except for those assigned tests where the <code>recordVideo</code> is set.</td> </tr> <tr> <td>Never</td> <td>Video recording is always disabled, even for those assigned tests where the <code>recordVideo</code> is set.</td> </tr> <tr> <td>Always</td> <td>Records a video on every execution of the execution plan.</td> </tr> <tr> <td>On Error</td> <td>Records a video when an error occurs during the execution of the execution plan.</td> </tr> </tbody> </table> <p>For additional information, see Video Capturing.</p> | Value | Description | --- | Video recording is disabled, except for those assigned tests where the <code>recordVideo</code> is set. | Never | Video recording is always disabled, even for those assigned tests where the <code>recordVideo</code> is set. | Always | Records a video on every execution of the execution plan. | On Error | Records a video when an error occurs during the execution of the execution plan. |
| Value | Description | | | | | | | | | | |
| --- | Video recording is disabled, except for those assigned tests where the <code>recordVideo</code> is set. | | | | | | | | | | |
| Never | Video recording is always disabled, even for those assigned tests where the <code>recordVideo</code> is set. | | | | | | | | | | |
| Always | Records a video on every execution of the execution plan. | | | | | | | | | | |
| On Error | Records a video when an error occurs during the execution of the execution plan. | | | | | | | | | | |

| Item | Description | | | | | | | | | | |
|---------------------------------------|---|-------|-------------|-----|--|-----------------|---|---------------------|--|---------------------------------|---|
| Screen Capturing | <p>Defines whether a screenshot is captured during the execution of the execution plan. This setting overrides the screen capturing settings of the individual tests that are assigned to the execution plan. You can select one of the following values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>---</td> <td>Screen capturing is disabled, except for those assigned tests where the <code>captureScreen</code> parameter is set.</td> </tr> <tr> <td>Never</td> <td>Screen recording is always disabled, even for those assigned tests where the <code>captureScreen</code> parameter is set.</td> </tr> <tr> <td>On Error</td> <td>Captures a screenshot when an error occurs during the execution of the execution plan.</td> </tr> </tbody> </table> <p>For additional information, see Screen Capturing.</p> | Value | Description | --- | Screen capturing is disabled, except for those assigned tests where the <code>captureScreen</code> parameter is set. | Never | Screen recording is always disabled, even for those assigned tests where the <code>captureScreen</code> parameter is set. | On Error | Captures a screenshot when an error occurs during the execution of the execution plan. | | |
| Value | Description | | | | | | | | | | |
| --- | Screen capturing is disabled, except for those assigned tests where the <code>captureScreen</code> parameter is set. | | | | | | | | | | |
| Never | Screen recording is always disabled, even for those assigned tests where the <code>captureScreen</code> parameter is set. | | | | | | | | | | |
| On Error | Captures a screenshot when an error occurs during the execution of the execution plan. | | | | | | | | | | |
| Execution Stop Condition | <p>Defines whether the execution stops if a test fails or if a test is not executed. This setting is especially useful if it makes no sense to keep a set of tests running if even one test fails. Use this setting to save resources and to be notified earlier if a test fails. You can select one of the following values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>---</td> <td>The execution proceeds, even if a test is not executed or fails.</td> </tr> <tr> <td>On Error</td> <td>The execution stops with the first failed test.</td> </tr> <tr> <td>Not Executed</td> <td>The execution stops with the first not executed test.</td> </tr> <tr> <td>On Error or Not Executed</td> <td>The execution stops with the first failed or not executed test.</td> </tr> </tbody> </table> <p> Note: If an execution is stopped, the results of the successfully executed tests are preserved. Test packages (with all subordinated tests) will always be executed to the end. They will not be stopped intermittently. However, data-driven tests will be stopped as soon as a subordinated test fails.</p> | Value | Description | --- | The execution proceeds, even if a test is not executed or fails. | On Error | The execution stops with the first failed test. | Not Executed | The execution stops with the first not executed test. | On Error or Not Executed | The execution stops with the first failed or not executed test. |
| Value | Description | | | | | | | | | | |
| --- | The execution proceeds, even if a test is not executed or fails. | | | | | | | | | | |
| On Error | The execution stops with the first failed test. | | | | | | | | | | |
| Not Executed | The execution stops with the first not executed test. | | | | | | | | | | |
| On Error or Not Executed | The execution stops with the first failed or not executed test. | | | | | | | | | | |
| Silk Test Classic AUT Hostname | <p>Lists all Silk Test Classic AUT hosts that are defined for this execution plan. Click Edit to edit the list of Silk Test Classic AUT hosts. For additional information, see Adding a Silk Test Classic AUT Host.</p> | | | | | | | | | | |

Mobile Device Selection

Lists the mobile device that has been selected to execute the selected execution plan. Click **Edit** to select a different device. Note that the device provider and your current project must be assigned to the same location, otherwise the devices will not show up in the list. For additional information, see [Managing Device Providers](#).

Code Analysis Settings

Details code-analysis settings that are defined for this execution plan. Click **Active/Inactive** to enable or disable code analysis for this execution plan. For additional information, see [Code Analysis](#).

Adding or Removing Manual Testers

For execution plans that include manual tests, the **Deployment** page enables you to assign users as manual testers to a selected execution plan. You can assign multiple manual testers to the execution plan.

To assign a manual tester to an execution plan or remove a tester from the selected execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan that you want to edit.
3. Click the **Deployment** tab.
4. Click **Edit** in the **Manual Test Execution** section. The **Add/Remove Testers** dialog box appears.
5. Select the user group you want to use from the **Available** list.
6. Add or remove testers:
 - Select one or more users in the **Available** list and click **Add** or **Add All**.
 - Select one or more users in the **Selected** list and click **Remove** or **Remove All**.

You can select multiple users with **Ctrl+Click** or **Shift+Click**.
7. Click **OK** to close the **Add/Remove Testers** dialog box.

Dynamic Hardware Provisioning with Keywords

The hardware-provisioning technology of Silk Central helps you manage test environments that include numerous execution servers. Rather than having to configure a one-to-one static execution-server assignment for each automated execution plan, keywords enable Silk Central to automatically select the most appropriate execution server for each execution plan. This is done through dynamic comparison of each execution plan's keyword list with the keyword lists of all active execution servers.

Keyword Concepts

Keywords typically describe your execution environment requirements, for example platform, operating system, and pre-installed applications. There are different uses for keywords, depending on whether the execution plan is executed automatically or manually.

| | |
|----------------------------------|--|
| Automated execution plans | When an automated execution plan is executed, Silk Central compares the execution plan's keywords with the keywords of all available execution servers. The execution is then run on the first-identified execution server that has a matching keyword list. |
| Manual execution plans | For manual execution plans, the manual tester can reflect the test environment by using keywords. |

If you require an automated execution plan to be run on multiple execution servers, create a copy of the execution plan and assign additional keywords to the execution plan that match other execution servers.

Reserved Keywords

If you do not require hardware provisioning to execute automated execution plans, you can use the reserved keywords that are created automatically for each execution server. In such cases, it is not necessary that you manually assign keywords to your execution servers. Instead, you can configure a one-to-one static execution server assignment for each execution plan.

A reserved keyword is assigned automatically to each newly created execution server. Reserved keywords are structured in the following form: `#<execution server name>@<location name>`.

Reserved keywords are only available when assigning keywords to execution plans.

In addition to the reserved keywords that are set up automatically for each defined execution server, reserved keywords are also set up for each execution server type:

| | |
|------------------|---|
| #PHYSICAL | Limits execution server provisioning to physical execution servers. |
| #VIRTUAL | Limits execution server provisioning to virtual execution servers. |

Keywords and Virtual Execution Servers

Keywords are assigned to virtual execution servers in the same way that they are assigned to physical execution servers. When you configure at least one virtual execution server, the **#VIRTUAL** keyword is

dynamically created and made available for assignment to all execution plans. If you prefer that an execution plan is executed on a virtual machine, select the **#VIRTUAL** keyword for the execution plan. When an execution plan has neither the **#VIRTUAL** nor **#PHYSICAL** keywords, the execution may occur on either a virtual or a physical execution server, assuming the settings of the execution environments are the same. When a test's keywords match multiple virtual execution servers, the first matching virtual execution server that is identified is selected.

Folder Execution

The execution plans can be combined into execution folders, where a folder can include execution subfolders and execution plans. The options for an execution plan execution are also available for an execution folder execution.

When executing a folder, the contained subfolders and execution plans are treated as follows:

| Keywords of Executed Folder | Keywords of Contained Execution Plan/Subfolder | Execution of Contained Execution Plan/Subfolder |
|-----------------------------|--|---|
| Has no keywords | Has no keywords | An execution plan without keywords obtain status NOT EXECUTED after execution |
| Has no keywords | Has keywords | Execution servers are assigned based on the execution plan/subfolder keywords |
| Has keywords | Has no keywords | Execution servers are assigned based on the folder keywords |
| Has keywords | Has keywords | Execution servers are assigned based on the folder keywords |

 **Note:** When a folder is executed manually and there are no keywords assigned, or no execution server exists for the assigned keywords, these execution plans are marked as **Not Executed**.

Creating New Keywords

To create new keywords:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan that you want to edit.
3. Click the **Deployment** tab.
4. In the **Execution Environment** section of the page, click **Edit**. The **Assign Keywords** dialog box appears. All keywords that have been defined for your execution environment are listed here.

 **Note:** The default reserved keywords for each execution server, `#<execution server name>@<location name>`, are included in the list.

5. Type an alphanumeric keyword into the **Keyword** field that describes the required environment for the execution plan.

For example, the keyword might be the name of the platform, operating system, or pre-installed applications. The following characters can not be used in keywords: `#$?*\\, ; ' "`.

 **Note:** Keywords are case insensitive. For example, "Vista" and "vista" are handled as the same keyword.

6. Press **Enter**.

The new keyword is now available for assignment.

Assigning Keywords to Execution Plans

To assign keywords to execution plans:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan that you want to edit.
3. Click the **Deployment** tab.
4. In the **Execution Environment** section of the page, click **Edit**. The **Assign Keywords** dialog box appears. All keywords that have been defined for your execution environment are listed here.



Note: The default reserved keywords for each execution server, #<execution server name>@<location name>, are included in the list.

5. Select a keyword in the **Select or enter keywords** list or directly enter a new keyword. Select multiple keywords with **Ctrl+Click** or **Shift+Click**.



Tip: The **Select or enter keywords** field is auto-complete enabled. When you enter alphanumeric characters, the field is dynamically updated with an existing keyword that matches the entered characters. The field is disabled when multiple keywords are selected in the **Select or enter keywords** or **Assigned keywords** lists.



Tip: If you only have a few execution servers and do not require hardware provisioning, it might be enough to use only the default, reserved keywords that are created for each execution server. In such cases, it is not necessary that you select additional keywords.

6. Click **>** to move the keyword into the **Assigned keywords** list. Click **<** to remove keywords from the list. You can also double-click keywords to move them from the one list to the other.



Note: The execution servers that match the assigned keywords are listed below in the dynamically-updated **Matching execution servers** list. This list updates each time you add or remove a keyword. Click on the name of an execution server in the list to access the execution servers in **Administration > Execution Servers**.

7. Click **OK** to save the keywords and close the **Assign Keywords** dialog box.

Removing Keywords from Execution Plans

To remove execution plan keyword assignments:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan that you want to edit.
3. Click the **Deployment** tab.
4. In the **Execution Environment** section of the page, click **Edit**. The **Assign Keywords** dialog box appears. All keywords that have been defined for your execution environment are listed here.



Note: The default reserved keywords for each execution server, #<execution server name>@<location name>, are included in the list.

5. Select the no longer needed keywords in the **Assigned keywords** list. Use **Ctrl+Click** or **Shift+Click** to select multiple keywords.
6. Click **<** to remove the keyword assignments.
7. Click **OK** to save the keywords and close the **Assign Keywords** dialog box.



Note: Keywords that are not in use anymore are automatically deleted from the system.

Exclusive Execution

Exclusive execution defines whether assigned tests of other execution plans having this setting disabled can be executed on the same matching execution server at the same time (**Disabled**), or if only tests of one single execution plan can run at the same time (**Enabled**). Disabling exclusive execution reduces the

required execution time, and is especially useful for mobile testing, where execution plans can execute on multiple devices on a device provider at the same time. No changes to your current execution configuration are required when disabling exclusive execution.

Before you enable parallel execution, be aware of the following:

- All Silk Central test types except Silk Performer support parallel execution.
- Your test scripts should not interfere with each other when being executed on the same computer at the same time. This means that they should not write data to a common folder, and the application under test must support the scripted actions in parallel.
- The **Common working folder** in the used source control profile must be empty. This enables Silk Central to use a temporary working folder.
- If an execution plan runs on an execution server, for which the exclusive execution setting is enabled, no other execution plans are executed in parallel, regardless of their setting.
- No video will be available if another execution plan is already recording during a parallel execution.
- For additional information on executing Silk Test tests in parallel, refer to the following documentation:
 - [Running Silk Test Workbench tests in parallel](#)
 - [Running Silk4J tests in parallel](#)
 - [Running Silk4NET tests in parallel](#)
 - [Running Silk Test Classic tests in parallel](#)

Distributed Execution

Distributed execution defines whether tests of an execution plan are executed sequentially on the selected execution server (**Disabled**), or if they are distributed to all matching execution servers and run in parallel (**Enabled**). No changes to your current execution configuration are required when turning on distributed execution.

Before you enable distributed execution, be aware of the following:

- If the individual tests of an execution plan have a very short execution time, enabling distributed execution will not improve the overall execution time, as the administrative overhead of test deployment, source control management, and result fetching will outweigh the benefits of executing the tests in parallel.
- All Silk Central test types support distributed execution.
- No code coverage information will be available if another test is already recording on the same execution server.
- No video will be available if another test is already recording on the same execution server.
- If one of the tests has an execution stop condition, all currently running and not yet started tests are stopped if this condition is met. Currently running tests are set to **Failed**, while not yet started tests are set to **Not Executed**. Tests that were already set to **Passed** before the condition was met will remain with that status.
- Virtual execution servers are not considered for distributed execution.

External Execution

Switching on external execution mode for an execution plan enables you to execute test suites in an external third-party system while maintaining the corresponding automated tests and results in Silk Central.

The external execution environment obtains test data and pushes back test results to Silk Central by using a REST API, for which an interactive documentation is available from [host:port\[/inst\]/Services1.0/swagger-ui.html](http://host:port[/inst]/Services1.0/swagger-ui.html).

Using REST API calls, clients can get pending execution plan runs and test details like keywords, attributes, and parameters. Additionally, clients can push results and result files to Silk Central after running these tests.

In external execution mode, no execution server is necessary to execute tests, unless there is an automated setup or cleanup test. A setup test for an external execution plan run is a custom script that

prepares the external execution environment and that initiates the external execution of the test suite. A cleanup test for an external execution plan run can be used to keep the execution server and execution environment reserved until the execution is finished. Automated setup and cleanup tests are always executed on a Silk Central execution server. You can use the environment variable `#sctm_regular_execdef_run_id` within a setup test to directly access the associated external execution plan run in your API requests.

Using a setup test increases the efficiency of your external execution plan runs. Without using a setup test, the external system can either frequently poll for pending execution plan runs, or directly start an execution plan through the API.

You can enable the external execution mode in the [Execution Deployment Page](#).

Selecting Mobile Devices for Automated Testing

Before you can select a mobile device (physical device, emulator or simulator) on which to run your execution plan, a device provider needs to be configured. Note that the device provider and your current project must be assigned to the same location, otherwise the devices will not show up in the list. For additional information, see [Managing Device Providers](#).

To select a mobile device:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan for which you want to select a mobile device.
3. Click the **Deployment** tab.
4. Click **Edit** in the **Mobile Device Selection** section. The **Select Device** dialog box appears.
5. Select a mobile device, device filter, or a native app:
 - To select a specific device, use the list on the right-hand side.
 - To select a device filter, select **Any device** in the list on the right-hand side. Edit this filter using the **Search** field, the **Platform** and the **Application** lists on the left-hand side.
 **Tip:** Use the search field to search for specific device types, for example by searching for `Emulator` or `Simulator`.
 - To install a native app on the mobile device for testing, select **Native App** in the **Application** list on the left-hand side and specify a URL or a UNC path to the app in the field below (iOS requires a URL). Example: `\\fileserver\apps\testapp.apk`
6. *Optional:* Check the **Use physical devices only** check box if you want to exclude emulators and simulators for your device selection.
7. *Optional:* Check the **Hide used or blocked devices** check box to only list devices that are currently available for use. This is especially useful if you want to execute a manual test, or if you want to run an execution plan immediately.
8. Click **OK** to save your mobile device selection.

During execution, either the selected device or the first matching device of a filter will be used and blocked until the execution plan has finished. If the selected device is not available or a filter does not match any available devices, the execution job stays in the queue until a suitable execution server and mobile device becomes available. Sauce Labs devices are not actually single devices, but rather device types, so they are not blocked during execution.

To change or delete your mobile device selection at a later point in time, use the **Edit** and **Delete** buttons, respectively.

Code Analysis

The goal of conventional code coverage is to deliver information about what code is covered by tests. This approach is typically used to gain code-coverage information for unit tests. The code-analysis functionality in Silk Central goes well beyond this, delivering data for unit, functional, and load tests, both automated and manual, in managed environments. Code coverage measurements are utilized to track test progress and

guide test planning. This innovative approach in Silk Central to code coverage draws on the relationship between specific tests and the code they test. This approach enables you to perform impact/dependency analysis of code changes from the testing perspective. It also assists you in optimizing your testing, by helping you identify the test runs that are most relevant to a specific code change.

The **Code Analysis** unit offers code-coverage data for AUT (Applications Under Test), packages/namespaces, classes, methods, and statements, enabling you to perform test-impact analysis, which determines the tests that should be run in response to specific code changes, and effort analysis, which determines how many hours of automated testing and manual testing are required to adequately cover specific code changes. Each of these code-analysis tasks can be addressed by running pre-configured reports. The unit features a navigation tree that lists all products that have been created for the selected project. You can drill down into products to select specific versions, and at the deepest level, specific builds.

Enabling Code Analysis

You must configure Silk Central to gather code coverage data from an application under test. You can configure any number of execution plans listed in **Executions > Details View > Deployment**.



Tip: Make sure that only code coverage-enabled tests access your application under test and that no parallel tests are executed. This ensures that the code coverage data is correct and that it can be assigned to a specific test later.

Java Code Analysis Options

To use Java Code Analysis for an AUT, you have to run the JVM of the AUT with a Java Agent. You can download the Java Agent package by selecting **Java Code Analysis Agent** under **Help > Tools** in the Silk Central menu. Java Code Analysis requires the latest version of the Java Runtime Environment (JRE) 1.8 to be installed. You can use the following options for all supported Java versions:

| Option | Description |
|---|--|
| <code>port=19129</code> | <i>Optional:</i> Port of code coverage service. When no port is specified the port 19129 is used. |
| <code>coveragepath=[path1], [path2], ...</code> | Pathes to the .jar files that are covered by the agent. If parts of a path contain whitespaces, you have to use quotes. For example: <code>coveragepath="C:\Program Files\YourApplication\"</code> . If a folder is specified, the agent recursively looks for all .class files in this folder. If a folder with a trailing wildcard is specified, the agent looks for .jar files in this folder. Running the agent without specifying a coveragepath is not valid. <pre>java -javaagent:agent.jar=coveragepath="C:Program Files\YourApplication\" Aut.jar</pre> |
| <code>includes=[package1]: [package2]: [package3]: ...</code> | <i>Optional:</i> Only packages that match these packages are considered for coverage. Wildcards (*) are allowed for specifying the included packages. <pre>java -javaagent:agent.jar=includes=com.borland.* AUT.jar</pre> |
| <code>excludes=[package1]: [package2]: [package3]: ...</code> | <i>Optional:</i> Packages that are specified in the excludes option are ignored for code coverage. Wildcards (*) are allowed for specifying the excluded packages. <pre>java -javaagent:agent.jar=includes=com.borland.*; excludes=com.borland.internal* AUT.jar</pre> |



Note: When both inclusions and exclusions are specified in the filters, the exclusions are evaluated first. If a class is both included and excluded through filters, this class will be excluded and no coverage data is generated for this class. If the filter option is not

Option

Description

specified, coverage data is generated for all used classes, unless the coverage path option is set, in which case coverage data is generated for all classes in the coverage path.

Java Code Analysis Example

Example for Windows:

```
"<java_home_directory>\bin\java" -  
javaagent:agent.jar=port=19129;  
coveragepath=C:\dev\deploy\lib\library1.jar,C:\dev\deploy\lib  
\library2.jar;  
includes=com.borland.*;excludes=com.borland.internal.*  
com.borland.MyApplication
```

Example for Linux (including quotation marks):

```
"<java_home_directory>/bin/java" -  
javaagent:agent.jar="port=19129;  
coveragepath=/tmp/dev/deploy/lib/library1.jar,/tmp/dev/  
deploy/lib/library2.jar;  
includes=com.borland.*;excludes=com.borland.internal.*"  
com.borland.MyApplication
```

.NET Code Analysis

Installing the Windows Code Analysis Framework and DevPartner Analytics

Before you can perform this task, make sure that Microsoft .NET Framework 3.5 is installed on the machine where the Code Analysis Framework will be installed (the machine that hosts the application to test). Older or newer versions of Microsoft .NET Framework will not work!

1. Go to the machine that hosts the application to test.
2. Open Silk Central.
3. Go to **Help > Tools** and click **Windows Code Analysis Framework**. This will prompt a download of `CAFramework.exe`. `CAFramework.exe` includes the installation for the Code Analysis Framework and the DevPartner Analytics package.



Note: Multiple execution servers running tests against the same Code Analysis Framework at the same time is not supported.

4. Double click the download file to start the installation. Walk-through the installation of the **Windows Code Analysis Framework**. At the end of the installation, you will be prompted to install `DevPartner Analytics`. Ensure that the **DevPartner Analytics Installation** check box is checked on the appropriate installation screen and install this component. At the end of this installation, all required files are installed and the proper service is started. The name of the service is `Silk Central Code Analysis Service`.

.NET Code Analysis and Manual Testing

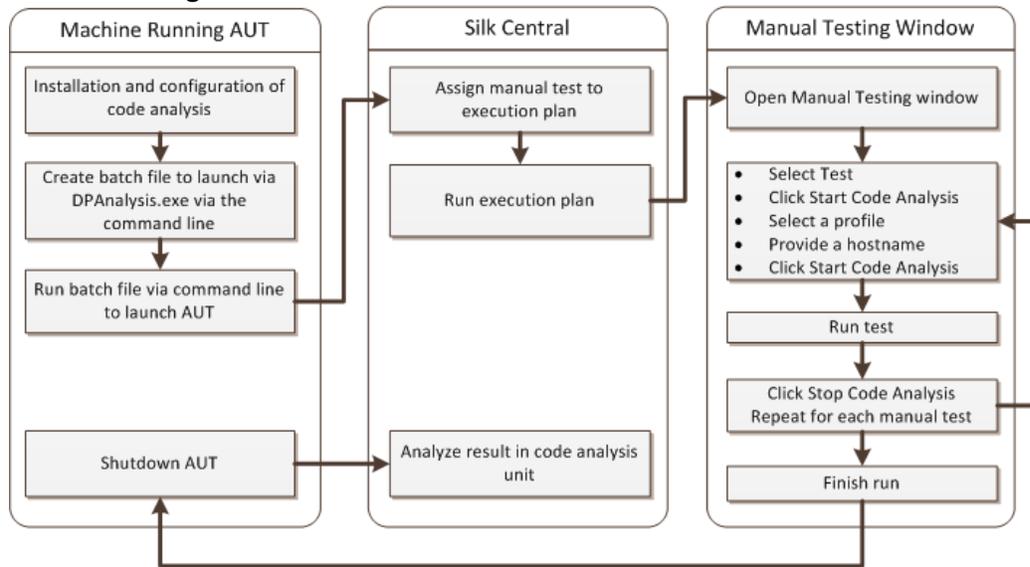
The following sections detail the required steps to install and use the DevPartner .NET code coverage components shipped with Silk Central. The example below will get you started by using manual testing and a single AUT. If you plan to use automated testing, first read and understand the steps below, and then see the topic titled *.NET Code Analysis and Automated Testing*.



Note: In order to use the Silk Central/.NET code coverage functionality, the Micro Focus DevPartner Analytics components version 11.2 or later must be used.

 **Note:** .NET code analysis requires the PDB files generated at compile time of the .NET application under test (AUT) to be located in the same directory as the AUT's executable(s). If the PDBs for the AUT are not available, it is not possible to leverage the .NET code analysis functionality.

Manual Testing Workflow



Configuring DPAnalysis to launch Your AUT

1. Create a .BAT file that points to DPAnalysis.exe. An example would be:

```
"C:\Program Files\Micro Focus\DevPartner\Analysis\dpanalysis.exe" /cov /
USE_AUTOMATION_INTERFACE
/W "C:\Common\Development\MYAPP"
/P "C:\Common\Development\MYAPP\MyApp.exe"
```

2. For readability, the example above is on multiple lines. Your .BAT file should be on one line. Make the following modifications:

- Ensure that the first line is your installation path to dpanalysis.exe. The two switches are mandatory.
- The second line contains a mandatory switch and needs to include the path to your executable and PDBs.
- The third line contains a mandatory switch and needs to include the absolute path to your test application and the application name.

3. Double-click the .BAT file to have DPAnalysis launch your application.

For more information on running DPAnalysis.exe, see the topic titled *Using DpAnalysis.exe*. That topic and related topics contain information on additional switches and how to use a configuration file if you have multiple applications to test.

Configuring Silk Central for Code Analysis

- In the menu, click **Execution Planning > Details View**.
- Open an existing execution plan that contains a manual test with steps to test functionality for your AUT.
- Click the **Deployment** tab.
- Under **Code Analysis Settings**, click **Edit** to view the **Edit Code Analysis Settings** dialog box.
- Ensure that **Enable Code Analysis** is checked.
- Select DevPartner .NET Code Analysis from the **Code Analysis Profile** list.

- For the `Hostnames` field, type in the host name followed by colon and the port number. For example: `localhost:19129`.

The default code analysis framework port is 19129. You can configure the code analysis framework to run at a given port using the following xml file: `<install dir>\Code Analysis \CodeAnalysisConfig.xml`

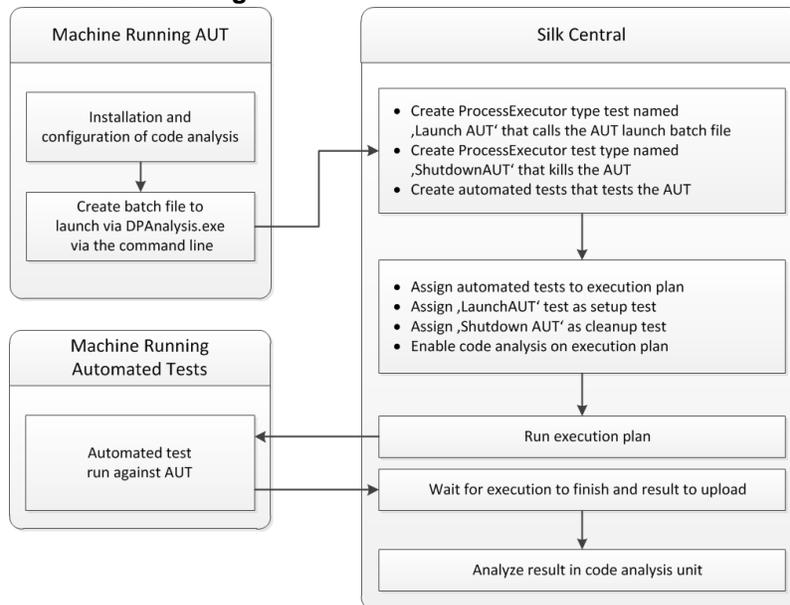
- Click **OK**.
- Run the test by clicking **Run** in the **Execution** area.
- On the **Run** dialog box, ensure that **All Tests** is checked.
- In the **Set build for execution plan**, select the proper build for your execution plan.
- Click **OK**.

.NET Code Analysis and Automated Testing

This section describes two scenarios that you may encounter when executing your tests that use code coverage. It also provides proper ways to structure your tests within an execution plan. The examples used in the following topics are specific to DevPartner Code Analysis, but the structure is similar for other code analysis tools. For example, you will have different startup/cleanup parameters in your execution plan.

 **Note:** There may be a delay between starting the AUT and starting code analysis, which may lead to some of the start-up code of the AUT not being recorded by code analysis.

Automated Testing Workflow



Configuring One or More Tests with One AUT

If you have the scenario where you have one or more tests in a single execution plan and they test against a single application under test, you should structure your testing as follows:

- Create a `ProcessExecutor` test that will start your code analysis application.
The `ProcessExecutor` **Argument List** should start your code analysis program.
For example:

- Executable Name:

```
cmd
```

- **Argument List:**

```
/c start cmd
/c D:\sctm\dps\startDPS_Simple1.bat
```

Where startDPS_Simple1.bat contains:

```
"C:\Program Files (x86)\Micro Focus\DevPartner\Analysis\dpanalysis.exe" /
cov /USE_AUTOMATION_INTERFACE /O "D:\SCTM\temp" /W "D:\SCTM\DPS\TESTAPP
\SCTMCodeAnalysisTestApp\bin\Debug" /P "D:\SCTM\DPS\TESTAPP
\SCTMCodeAnalysisTestApp\bin\Debug\SCTMCodeAnalysisTestApp.exe"
```

2. Create an execution plan.
3. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Setup Test** row.
4. Add the previously created test as the **Setup Test**.
5. Add a test to the execution plan that automates your application under test with an automated testing tool.
6. *Optional:* Add another test to the execution plan that automates your application under test with an automated testing tool.
7. Create a `ProcessExecutor` test.

The `ProcessExecutor` **Argument List** should close your program:

- **Executable Name:**

```
cmd
```

- **Argument List:**

```
/c TASKKILL /IM SCTMCodeAnalysisTestApp.exe /F
```

8. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Cleanup Test** row.
9. Add the previously created test as the **Cleanup Test**.

Configuring One or More Tests with Multiple AUTs

If you have the scenario where you have one or more tests in a single execution plan and they test against multiple AUTs, you should structure your testing as follows:

1. Create a `ProcessExecutor` test that will start your code analysis application.

The `ProcessExecutor` **Argument List** should start your code analysis program. For example:

- **Executable Name:**

```
cmd
```

- **Argument List:**

```
/c start cmd
/c D:\sctm\dps\two_app_dps.cmd
```

Where two_app_dps.cmd contains:

```
"C:\Program Files (x86)\Micro Focus\DevPartner\Analysis\dpanalysis.exe" /
config D:\SCTM\DPS\configuration_file.xml
```

and configuration_file.xml contains:

```
<?xml version="1.0" ?>
<ProductConfiguration>
  <RuntimeAnalysis Type="Coverage" MaximumSessionDuration="1000"
NoUIMsg="true" />
  <Targets RunInParallel="true">
    <Process CollectData="true" Spawn="true" NoWaitForCompletion="true"
NM_USE_AUTOMATION_INTERFACE="1" >
      <AnalysisOptions NM_USE_AUTOMATION_INTERFACE="1" NO_MACH5="1"
NM_METHOD_GRANULARITY="1" SESSION_DIR="c:\temp" />
    </Process>
  </Targets>
</ProductConfiguration>
```

```

        <Path>D:\SCTM\DPS\SCTMCodeAnalysisTestApp
\SCTMCodeAnalysisTestApp\
        SCTMCodeAnalysisTestApp\bin\Debug\SCTMCodeAnalysisTestApp.exe</
Path>
        <Arguments></Arguments>
        <WorkingDirectory>D:\SCTM\DPS\SCTMCodeAnalysisTestApp
\SCTMCodeAnalysisTestApp\
        SCTMCodeAnalysisTestApp\bin\Debug</WorkingDirectory>
    </Process>
    <Process CollectData="true" Spawn="true" NoWaitForCompletion="true"
>
        <AnalysisOptions NM_USE_AUTOMATION_INTERFACE="1" NO_MACH5="1"
NM_METHOD_GRANULARITY="1" SESSION_DIR="c:\temp" />
        <Path>D:\SourceCode\SVNQAD\Development\Development\SOURCECODE\
QADSCTMMigration\bin\QADSCTMMigration.exe</Path>
        <Arguments></Arguments>
        <WorkingDirectory>D:\SourceCode\SVNQAD\Development\Development\
SOURCECODE\QADSCTMMigration\bin</WorkingDirectory>
    </Process>
</Targets>
</ProductConfiguration>

```

2. Create an execution plan.
3. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Setup Test** row.
4. Add the previously created test as the **Setup Test**.
5. Add a test to the execution plan that automates your first application under test with an automated testing tool.
6. Add another test to the execution plan that automates your second application under test with an automated testing tool.
7. Create a `ProcessExecutor` test.

The `ProcessExecutor` **Argument List** should close your programs:

- Executable Name:

```
cmd
```

- Argument List:

```
/c TASKKILL /IM application1.exe /IM Application2.exe /F
```

8. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Cleanup Test** row.
9. Add the previously created test as the **Cleanup Test**.

Using DPAnalysis.exe

`DPAnalysis.exe` is the DevPartner code coverage application that allows you to profile .NET applications. The application is started using the command line.

You can use the application with traditional switches as documented in these topics, or you can add an additional switch that points to a configuration file. You may want to use a configuration file if you have an exceptionally long command line string or you want to run multiple applications in a single session.

The `/USE_AUTOMATION_INTERFACE` switch is mandatory. It causes the automation interface environment needed for the interaction with the framework to be established. It also causes license checking to be bypassed. This switch does not take any parameters.

Running DPAnalysis.exe from the Command Line

The DevPartner Studio Code Coverage installation includes `DPAnalysis.exe`, a command line executable that is installed in the `\Program Files (x86)\Micro Focus\DevPartner\Analysis\` directory.

Use the following syntax and switches to run the DevPartner Studio Code Coverage tool from the command line:

```
DPAnalysis [/Cov] [/USE_AUTOMATION_INTERFACE] [/E|/D|/R]
[/W workingdirectory] [/PROJ_DIR] [/H hostmachine]
[/NOWAIT] [/NO_UI_MSG] [/N] [/F] [/A C:\templ;C:\temp2]
[/NO_QUANTUM /NM_METHOD_GRANULARITY /EXCLUDE_SYSTEM_DLLS
/NM_ALLOW_INLINING /NO_OLEHOOKS /NM_TRACK_SYSTEM_OBJECTS]
{/P|/S} target [target arguments]
```

For detailed information on the available switches and about how to use a DPAnalysis configuration file, refer to the DevPartner documentation.



Note: DPAnalysis.exe does not instrument unmanaged code. To collect performance or coverage analysis data for an unmanaged application, you must first instrument the application.

Example

You can use DPAnalysis.exe directly from the command line, using switches to direct the analysis session. For example, the following command line launches a code coverage session for the application SCTMCodeAnalysisTestApp.exe.

```
dpanalysis.exe /cov /USE_AUTOMATION_INTERFACE /W "D:\SCTM
\DPSCoverageIntegration\Test App\
SCTMCodeAnalysisTestApp\SCTMCodeAnalysisTestApp\bin\Debug" /P
"D:\SCTM\DPSCoverageIntegration\Test App\SCTMCodeAnalysisTestApp
\SCTMCodeAnalysisTestApp\bin
\Debug\SCTMCodeAnalysisTestApp.exe"
```

Enabling Code Analysis for Execution Plans

To enable code analysis for an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, select an execution plan.
3. Click the **Deployment** tab.
4. In the **Code Analysis Settings** section of the **Deployment** page, click **Edit**. The **Edit Code Analysis Settings** dialog box displays.
5. Check the **Enable code analysis** check box.
6. Select a profile from the **Code Analysis Profile** list box.
7. In the **Hostnames** field, enter a comma-separated list of host names, with port, if default port 19129 is not used, from which code analysis information is to be gathered.

For example `labmachine1, labmachine2:8000, 198.68.0.1`. For each execution plan, you need to define the host names of the machine resources where the AUT is running. For example, with a client/server system, you must not only gather code coverage information on the client, which probably runs directly on an execution server, but also on the server, which likely runs on a different machine. This applies to all multi-tiered applications.



Note: For JUnit code analysis runs, you do not need to specify a hostname.

8. Click **OK** to save your settings.



Note: Once code analysis has been defined for an execution plan, each future run of that execution plan will gather code coverage information from the defined host names. While monitoring an execution on the **Activities** page, you will see that after gathering the sources for tests, Silk Central gathers full code coverage information before beginning test runs. The Code Coverage Controller, which is integrated into each Silk Central execution server, controls all defined hosts during execution runs. For each test of an execution plan, the controller starts and stops all associated instances,

collects XML-based code coverage files for the test, and merges the results into a single file. The test then saves the merged code coverage file to its execution results.

Code Analysis Details Page

Reports > Code Analysis > Details

The **Details** page displays code-coverage information for selected products, versions, and builds at the product, package, and class levels.

Product level view displays a list of covered and not-covered packages for specific products and product builds. By clicking a package name in **Product** view you can drill down to view code-coverage information for the classes that are included in that package.

Product view displays the following attributes for a selected product:

| Attribute | Description |
|---------------------|--|
| Name | Product name |
| Statements | Total amount of statements |
| Packages/Namespaces | Histogram bar view that includes the following figures: <ul style="list-style-type: none"> Total percentage of packages/namespaces that are covered Number of covered packages/namespaces, in green Number of uncovered packages/namespaces, in red |
| Classes | Histogram bar view that includes the following figures: <ul style="list-style-type: none"> Total percentage of classes that are covered Number of covered classes, in green Number of uncovered classes, in red |
| Methods | Histogram bar view that includes the following figures: <ul style="list-style-type: none"> Total percentage of methods that are covered Number of covered methods, in green Number of uncovered methods, in red |

Package level view displays a list of covered and not-covered classes for specific products and product builds. By clicking a class name in **Package** view you can drill down to view code-coverage information for the methods that are included in that class.

Package view displays the following attributes for each package in the selected product, across multiple rows:

| Attribute | Description |
|------------|--|
| Package | Package name |
| Statements | Total statements |
| Classes | Histogram bar view that includes the following figures: <ul style="list-style-type: none"> Total percentage of classes that are covered Number of covered classes, in green Number of uncovered classes, in red |
| Methods | Histogram bar view that includes the following figures: <ul style="list-style-type: none"> Total percentage of methods that are covered |

| Attribute | Description |
|-----------|--|
| | <ul style="list-style-type: none"> Number of covered methods, in green Number of uncovered methods, in red |

Class level view displays a list of covered and not-covered methods for specific products and product builds.

Class view displays the following attributes for each method, across multiple rows:

| Attribute | Description |
|------------|---|
| Name | Method name |
| Signature | Method signature |
| Statements | Total statements |
| Covered | Covered status of method: <ul style="list-style-type: none"> True indicates that the method is covered False indicates that the method is not covered |

 **Note:** When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

Latest Builds and Build Versions

When you select a product in the navigation tree, the list of packages and classes with coverage information for the latest covered version, which implies the latest covered build for the version, is displayed automatically. When you select a product version in the navigation tree, coverage information for the latest covered build of the version is displayed automatically.

 **Note:** If you are running multiple applications against the same build, the coverage from all of the applications will be merged together.

 **Note:** Code analysis across a range of builds is not supported by Silk Central currently.

Generating Code-Change Impact Reports

To generate a code-change impact report:

1. In the menu, click **Projects > Project List**.
2. Select the project for which you want to analyze code-coverage data.
3. In the menu, click **Reports > Code Analysis**.
4. Click **Create Code Change Impact Report**. The **Select Classes for Report** dialog box appears.
5. Select a **Product** and **Version** if you want to change the pre-selected values.
6. In the **Filter** field, type criteria to filter the packages/namespaces. For example, entering the string `published` will only list packages/namespaces that contain the string `published` in their names.
7. Select a package from the **Packages/Namespace** list.
Use **Ctrl+Click** or **Shift+Click** to select multiple packages/Namespace.
The classes that are available in the selected package/namespaces are displayed in the **Classes** list.
8. In the **Classes** list, select a class that you want to have included as a source in your report.
Use **Ctrl+Click** or **Shift+Click** to select multiple classes.
9. Click **Add**. The selected classes are added to the **Selected classes** list.

10. Repeat the preceding steps until you have added all required classes to the **Selected classes** list.

You can remove classes from the **Selected classes** list by selecting entries and clicking **Remove** or by clicking **Remove All**.

11. Select a report from the **Select report** list.

12. Click **OK** to generate the report.

Select Classes for Report Dialog Box

Reports > Code Analysis > Create Code Change Impact Report

The **Select Classes for Report** dialog box enables you to select class files to be included as sources in a *Code-Change Impact Report*. The dialog box includes the following elements:

| Element | Description |
|----------------------------|---|
| Product | Name of the product for which code analysis information is required. |
| Version | Version of the product for which code analysis information is required. |
| Filter | Enter criteria for filtering the packages. For example, if you type the string <code>published</code> , the <i>Code-Change Impact Report</i> displays only packages that contain the string “published” in their names. |
| Packages/Namespaces | Product packages/namespaces that are specified for coverage in the selected version. |
| Classes | All classes included in the selected package. |
| Add | Click to add the selected classes for code-coverage analysis. |
| Selected Classes | Displays the classes that are already selected for code-coverage analysis. |
| Remove | Click to remove the selected class from the Selected Classes list box. |
| Remove All | Click to remove all classes from the Selected Classes list box. |
| Select Report | Type of the <i>Code-Change Impact Report</i> . |

Viewing Code-Coverage Information for Packages/Namespaces

To view code-coverage information for a package/namespace:

1. In the menu, click **Projects > Project List**.
2. Select the project for which you want to view code-coverage information.
3. In the menu, click **Reports > Code Analysis**.
4. In the **Code Analysis** tree, expand the project node in the navigation tree to display the products that are available for the selected project.
5. Expand a product node to display the versions that are available for that product.
6. Expand a version node to display the builds that are available for that version.
7. Select a specific build. Code coverage information for the selected build displays on the **Details** page.

 **Note:** To view code-analysis information for all products click **Show all products** on the toolbar. Products of other projects are then listed under the **Other Projects** node.

Adding a Silk Test Classic AUT Host

For execution plans that run Silk Test Classic tests, you may have a setup where the Silk Test Classic agent is on a different computer than the execution server. In this case, you can define the location of the Silk Test Classic agent, the Silk Test Classic AUT (Agent Under Test) Hostname.

To add a a Silk Test Classic AUT host to the selected execution plan or edit the host:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan to which you want to assign the Silk Test Classic AUT host.
3. Click the **Deployment** tab.
4. In the **Silk Test Classic AUT Hostname** section, click **Edit**. The **Edit Silk Test Classic AUT Hostname** dialog box displays.
5. In the **Hostname** field, type the name of the computer where the Silk Test Classic agent runs.
Proper configuration of option files is required. For details on the command-line option `-m`, refer to the Silk Test Classic documentation.
6. Click **OK** to add the Silk Test Classic AUT host to the selected execution plan.

Configuring Execution Dependencies

An execution dependency allows you to configure the automatic execution of one execution plan based on the results of another execution plan. For example, *If execution plan 'A' fails, automatically execute execution plan 'B'*.

Adding Dependent Execution Plans

To add a dependent execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan that will act as the master execution plan.
3. Click the **Dependencies** tab.
4. Click **Add dependent Execution Plan**. The **Add dependent Execution Plan** dialog box appears.
5. From the **Condition** list, select the condition that is to trigger the dependent execution plan.
 - Any
 - Passed
 - Failed
 - Not Executed

The **Any** status means that the dependent test execution will trigger no matter what the status of the previous test execution is.

6. Select an execution plan from the **Execution Plans** tree.
7. Specify where the dependent execution plan is to be deployed.

| Option | Description |
|---|---|
| As specified in the dependent Execution Plan | Automated tests assigned to the dependent execution plan will be executed on the execution server specified for the dependent execution plan on the Deployment page. Manual tests assigned to the dependent execution plan will be assigned to the users specified for the dependent execution plan on the Deployment page. |
| Same as <selected execution plan's execution server> | Automated tests assigned to the dependent execution plan will be executed on the execution server specified for the master execution plan on the Deployment page. Manual tests assigned to the dependent execution plan will be assigned to the users specified for the master execution plan on the Deployment page. |
| Specific: Execution Server/ Manual Tester | Select a pre-configured execution server and/or a manual tester from the list boxes. Automated tests assigned to the dependent execution plan will be executed on the specified execution server. Manual tests assigned to the dependent execution plan will be assigned to the specified manual tester. If only a specific manual tester is defined and no server, only manual tests will be |

Option

Description

executed. If only a specific execution server is defined and no manual tester, only automated tests will be executed.

8. Click **OK** to create the dependency.



Note: Silk Central will not allow you to create cyclical execution dependencies. You can select conditions to fulfill for manual tests. For example, if the selected condition is `Failed` and all manual tests passed, but some automated tests failed, only automated tests assigned to the dependent execution plan are executed.

Editing a Dependency



Note: To edit an existing dependency, you must select the master execution plan, which is the definition for which a specific condition will trigger the execution of the dependent execution plan. You cannot edit dependency settings from the dependent execution plan.

To edit a previously configured dependency:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan that you want to edit.
3. Click the **Dependencies** tab.
4. In the **Dependent Execution Plans** section, click  in the **Actions** column of the dependent execution plan. The **Edit Dependency** dialog box opens.
5. From the **Condition** list, select the condition that is to trigger the dependent execution plan.
 - Any
 - Passed
 - Failed
 - Not Executed

The `Any` status means that the dependent test execution will trigger no matter what the status of the previous test execution is.

6. Specify where the dependent execution plan is to be deployed.

Option

Description

As specified in the dependent Execution Plan

Automated tests assigned to the dependent execution plan will be executed on the execution server specified for the dependent execution plan on the **Deployment** page. Manual tests assigned to the dependent execution plan will be assigned to the users specified for the dependent execution plan on the **Deployment** page.

Same as <selected execution plan's execution server>

Automated tests assigned to the dependent execution plan will be executed on the execution server specified for the master execution plan on the **Deployment** page. Manual tests assigned to the dependent execution plan will be assigned to the users specified for the master execution plan on the **Deployment** page.

Specific: Execution Server/ Manual Tester

Select a pre-configured execution server and/or a manual tester from the list boxes. Automated tests assigned to the dependent execution plan will be executed on the specified execution server. Manual tests assigned to the dependent execution plan will be assigned to the specified manual tester. If only a specific manual tester is defined and no server, only manual tests will be executed. If only a specific execution server is defined and no manual tester, only automated tests will be executed.

Deleting a Dependency

To delete a dependency:

1. In the menu, click **Execution Planning > Details View**.
2. Select the master execution plan from which you want to delete a dependency.
3. Click the **Dependencies** tab.
4. In the **Dependent Execution Plans** section, click  in the **Actions** column of the dependent execution plan.
5. Click **Yes** on the **Delete Dependency** dialog box to delete the dependency.

Execution Dependencies Page

Execution Planning > Details View > <Execution Plan> > Dependencies



Note: This page is not displayed for execution plans that are assigned to a testing cycle.

The **Dependencies** page lists dependent execution plans and master execution plans of the selected execution plan.

The page is divided into two parts. The **Master Execution Plans** are the execution plans for which a specific condition triggers the selected execution plan. The **Dependent Execution Plans** are the execution plans that will be triggered if the selected execution plan results in a specific condition.

Master Execution Plans

For each execution plan in the list, the page displays the following columns:

| Column | Description |
|------------------|--|
| Name | Name of the master execution plan that the selected execution plan is dependent upon. |
| Condition | Condition of the master execution plan that must be met for the selected execution plan to be triggered. |

Dependent Execution Plans

For each execution plan in the list, the page displays the following columns:

| Column | Description |
|-----------------------------------|---|
| Name | Name of the dependent execution plan that the selected execution plan serves as the master of. |
| Condition | Condition of the selected execution plan that must be met for the dependent execution plan to be triggered. |
| Execution Server / User(s) | Execution server where the dependent execution plan is to be run or, in the case of a manual test execution, manual tester who is to perform the manual test. |
| Actions | Actions that can be performed on the selected dependency. Edit Settings and Delete Dependency . |

Assigned Tests

This section describes how to assign tests to execution plans.

Assigning Tests from Grid View to Execution Plans

The tests that are assigned to the selected execution plans are listed on the **Assigned Tests** page.

To assign one or more tests from **Grid View** to one or more execution plans:

1. In the menu, click **Tests > Grid View**.
2. Select the tests you want to assign to execution plans.
You can select multiple tests with **Ctrl+Click** or **Shift+Click**.
3. Right-click the selected tests and click **Save Selection**.
4. In the menu, click **Execution Planning > Details View**.
5. Select the execution plan to which you want to assign the selected tests.
6. Click the **Assigned Tests** tab.
7. Click **Assign Saved Selection**.



Note: Only tests that reside in the test container of the execution plan are inserted. You can insert the selected tests to more than one execution plans. You can not insert them into requirements in a different project. The selection persists until you make a different selection or close Silk Central.

Manually Assigning Tests to Execution Plans

The tests that are assigned to the selected execution plans are listed on the **Assigned Tests** page.

To manually assign tests to an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan to which you want to assign the selected tests.
3. Click the **Assigned Tests** tab.
4. Click the **Manual assignment** option. All tests of the test container which is associated with the selected execution are displayed in the **Tests** tree. If you have created a test filter, you can select it from the filter list above the **Tests** tree. To create a new test filter, click **Tests** in the menu and click  (**New Filter**) in the toolbar.
5. Select the tests that you want to assign.
You can use your keyboard's **Ctrl** and **Shift** keys to select multiple tests using standard browser multi-select functions.
6. Drag your selection and drop it into the **Assigned Tests** area, or click  left of any test.
Assigning a folder or the top-level container assigns all child tests of that parent to the selected execution plan.



Tip: Right-click a node to expand and collapse it.

Assigning Tests to Execution Plans through a Filter

The tests that are assigned to the selected execution plans are listed on the **Assigned Tests** page.

You have to create a filter with the category **Test** before you can perform the following steps. See *Creating Filters* for details. Alternatively select an existing filter.

To use a filter to assign one or more tests to an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan to which you want to assign tests.
3. Click the **Assigned Tests** tab.
4. Click the **Assignment by filter** option.
5. Choose a filter from the list.



Note: When you switch from assigning tests by filters to manual assignment, the assigned tests remain.

Locating Tests Assigned to Execution Plans

To locate manually assigned tests in the **Tests** tree:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan for which you want to locate the assigned tests.
3. Click the **Assigned Tests** tab.
4. *Optional:* Select **Manual assignment**, if it is not selected.
5. In the **Actions** column of a test, click  to find out in which test folder or container the test is stored in.

The corresponding parent folder is expanded in the **Tests** tree and the assigned test is highlighted in blue.

Removing Test Assignments

To remove a manually assigned test:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan from which you want to remove the assignment.
3. Click the **Assigned Tests** tab.
4. In the **Actions** column of the assigned test, click , or use **Ctrl+Click** or **Shift+Click** to select multiple tests and press **Del**.

 **Tip:** To remove all assigned tests, click **Remove All**.

Execution Assigned Tests Page

Execution Planning > Details View > <Execution Element> > Assigned Tests

 **Note:** This page is not displayed for execution plans that are assigned to a testing cycle.

The **Assigned Tests** page lists all tests that are assigned to the selected execution plan or configuration suite. Use this page to assign additional tests to the execution plan or configuration suite, to remove tests from the execution plan or configuration suite, or to change the execution order of the assigned tests.

If you have created a test filter, you can select it from the filter list above the **Tests** tree. To create a new test filter, navigate to the **Tests** area and click **New Filter** on the toolbar.

 **Note:** All changes in this page are immediately applied.

 **Note:** When you access the page from a sub-element of a configuration suite, the page is read-only.

The page includes the following items:

| Item | Description |
|-------------------------------|--|
| Manual assignment | Click to manually assign tests to the execution plan or configuration suite. |
| Use tree order | Check to set the execution order of the assigned tests to follow the execution order in the Tests area. |
| Assign Saved Selection | Click to assign a selection of tests from Tests > Grid View . |
| Assignment by filter | Click to automatically assign tests to the execution plan or configuration suite based on a predefined filter. The available filters are listed in the list box. |
| Assigned Tests | Amount of tests that are assigned to the execution plan or configuration suite. |

| Item | Description | | | | | | |
|---|---|---------------|--------------------|---|---|---|--|
| Time Left in Testing Cycle [hh:mm] | <p>This field is available when an execution plan is in a testing cycle. It indicates the hours remaining in a testing cycle after subtracting the combined planned time of all tests from the Available Time in the testing cycle.</p> <p> Note: Planned Time is for manual tests only.</p> | | | | | | |
| Planned Time | <p>The estimated time to execute all tests in the execution plan or configuration suite. For configuration suites, the Planned Time is the sum of the planned times of all the configurations in the suite.</p> <p> Note: Planned Time is for manual tests only.</p> | | | | | | |
| Assigned Tests List | <p>For each assigned test, the page displays the following columns:</p> <p>Order The execution order of the test. Uncheck the Use tree order check box to change the execution order. Click in the field, type the new order of the test, and then press Enter to confirm the change. Each change in each of the fields must be confirmed by pressing Enter. If you change the orders of multiple tests without pressing Enter each time, just the last change before pressing Enter is taken.</p> <p>Actions You can perform the following actions on the assigned tests when the Manual assignment option button is clicked:</p> <table border="0" data-bbox="781 982 1333 1125"> <thead> <tr> <th data-bbox="781 982 906 1014">Action</th> <th data-bbox="906 982 1049 1014">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="781 1035 906 1077"></td> <td data-bbox="906 1035 1333 1077">Remove the selected test from the list.</td> </tr> <tr> <td data-bbox="781 1098 906 1125"></td> <td data-bbox="906 1098 1333 1125">Locate the selected test in the test tree.</td> </tr> </tbody> </table> <p>Test Name of the test. Click to access the test in the Tests area.</p> <p>Workflow State Current workflow state of the test. For more information, see <i>Test Asset Workflow</i>. This column is hidden by default.</p> <p>Status Status of the last run of the test in the context of the execution plan or the configuration suite. When the test is executed outside of the context of the execution plan or configuration suite, the displayed status remains unchanged. If the test has not yet been executed in the context of the execution plan or configuration suite, the status is N/A.</p> <p>For tests or test packages that are included in a configuration suite, the status is an aggregation of all statuses of all configurations within the suite:</p> <ul style="list-style-type: none"> • If the test is assigned only to the configuration suite, the status is Passed only if it is passed in all configurations. If the status is not Passed in all configurations, the status of the worst run is propagated, with the following top-down priority: <ol style="list-style-type: none"> 1. N/A 2. Not Executed 3. Failed | Action | Description |  | Remove the selected test from the list. |  | Locate the selected test in the test tree. |
| Action | Description | | | | | | |
|  | Remove the selected test from the list. | | | | | | |
|  | Locate the selected test in the test tree. | | | | | | |

| Item | Description |
|-------------|--|
| | <p data-bbox="818 239 935 264">4. Passed</p> <p data-bbox="818 291 1419 384">The status of a requirement that is assigned to the test is <code>Failed</code> if the test fails in one or more of the configurations.</p> <ul data-bbox="781 394 1442 674" style="list-style-type: none"> <li data-bbox="781 394 1442 575">• We do not recommend assigning the test to a configuration suite and one or more simple execution plans. However, in this case, the aggregate of the configuration statuses is one status, and the last statuses of the execution plans are other statuses. The status that was executed last determines the last status of the test. <li data-bbox="781 590 1442 674">• If the test is assigned to a simple execution plan only, the status is updated in response to the status of the last test run. <p data-bbox="626 709 740 764">Last Execution</p> <p data-bbox="781 709 1446 831">Date and time of the last run of the test in the context of the execution plan or configuration suite. When the test is executed outside of the context of the execution plan or configuration suite, the displayed time and date remain unchanged.</p> <p data-bbox="626 852 721 907">Planned Time</p> <p data-bbox="781 852 1281 877">The estimated time for the execution of the test.</p> <p data-bbox="781 898 1325 924"> Note: Planned Time is for manual tests only.</p> |
| Test | <p data-bbox="618 993 1442 1152">Shows all tests in the Tests tree that are available for assignment to the selected execution plan or configuration suite. Double-click a test or use  to assign the tests to the execution plan or configuration suite. For information about inserting multiple tests from the Tests area to the execution plan or configuration suite, see <i>Assigning Tests from Grid View to Execution Plans</i>.</p> |

Schedules

In the **Schedule** page, once you have defined the tests that are to be included in an execution plan, a folder, or a configuration suite, you can define the schedule, by which the execution plan, folder, or configuration suite is to be executed.

Three scheduling options are available:

- **None**
- **Global**
- **Custom**

 **Note:** You can define schedules for execution plans, folders, or configuration suites. If a schedule is defined for a folder or a configuration suite, all execution plans that are included in the selected folder or configuration suite will be executed at the specified schedule. All execution plans, folders, or configuration suites with no keywords assigned get the status `Not Executed` when executed in a schedule. The scheduling options for parts of a project copy or baseline are set to none to prevent losing the last execution status. For more information on copies and baselines of projects, see *Project Baselines* and *Copying Projects or Project Baselines*.

Definite Runs

Definite runs enable you to define times at which tests will be executed regardless of configured schedules.

Creating a Custom Schedule

To create a custom schedule for a selected execution plan, folder, or configuration suite:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite for which you want to configure a custom schedule.



Note: To save an edited version of a global schedule as a custom schedule, click **Edit** while the global schedule is selected in the list box. This enables you to edit the global schedule and save the result as a custom schedule.

3. Click the **Schedule** tab.
4. Click the **Custom** option to enable the scheduling controls.
5. Click **Edit**.
6. Click  next to the **From** field and use the calendar tool to specify the time and date when the execution schedule should begin.
7. Specify the **Interval** at which the tests should be executed.
8. In the **Run** section, specify when the schedule should end.
Select one of the following options:
 - Click **Forever** to define a schedule with no end.
 - Click **n Time(s)**.
 - Click  next to the **until** field and use the calendar tool to specify the time and date when the execution schedule should end.
9. *Optional:* Click **Add Exclusion** to define times when scheduled elements should not be executed.
10. *Optional:* Click **Add Definite Run** to define times when unscheduled executions should be executed.
11. Click **Save** to save your custom schedule.

Specifying Global Schedules

Silk Central offers the possibility of defining global schedules, which can be reused in Silk Central for the scheduling of tests. Global schedules can speed up the process of scheduling tests, since the need to define individual schedules for each test is reduced to only those tests that require special scheduling.

To select a predefined schedule that is globally available throughout Silk Central:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite for which you want to configure a schedule.
3. Click the **Schedule** tab.
4. Click the **Global** option button.
5. Select the required predefined schedule from the list box.

Details of the predefined schedule are displayed in a read-only calendar view. To save an edited version of a global schedule as a custom schedule, click **Edit**.



Note: You can configure global schedules in **Administration > Schedules**.

Specifying No Schedule

To specify that no schedule should be defined for an execution plan, a folder, or a configuration suite:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite, for which you want to define that it is not to be executed based on schedules.

3. Click the **Schedule** tab.
4. Click the **None** option button.

Schedule Exclusions

A schedule exclusion is a regularly occurring time period during which executions should be suspended, for example during a weekly planned system downtime or during weekends. You can add as many exclusions to a schedule as you need. Exclusions enable you to define weekdays and time-of-day intervals during which tests are not to be executed, regardless of configured schedules. For example, you may not want tests to be executed on weekends.

Adding Exclusions



Note: You must have administrator rights to edit global schedules. To define a scheduling exclusion for a global schedule, navigate to **Administration > Schedules**.

To add an exclusion to a custom schedule:

1. In the menu, click **Execution Planning > Details View**.
2. Select an execution plan, folder, or configuration suite, for which you want to add a scheduling exclusion.
3. Click the **Schedule** tab.
4. Click the **Custom** option to enable the scheduling controls.
5. Click **Add Exclusion**.
6. On the **Configure Schedule Exclusion** page, select the weekdays on which tests should be suppressed.
7. Define the specific time intervals on those days during which execution should be suppressed.
8. Click **OK**. Your exclusion settings are now listed on the **Schedule** page.
9. Click **Save** to add the exclusion to the current schedule, or continue adding additional exclusions.

Editing Exclusions

To edit an exclusion:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite, for which you want to edit a previously configured exclusion.
3. Click the **Schedule** tab.
4. In the **Actions** column of the exclusion, click .
5. Edit the exclusion as required and click **OK**.
6. Click **Save**.

Deleting Exclusions

To delete an exclusion:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite, for which you want to delete a previously configured exclusion.
3. Click the **Schedule** tab.
4. In the **Actions** column of the exclusion, click .

Definite Runs

A definite run is a run of an execution plan, a folder, or a configuration suite, that you schedule to run at a specific time, independent of the schedule that is configured for the execution plan, folder, or configuration suite. You can add as many definite runs to a schedule as you need.

Adding Definite Runs



Note: You must have administrator rights to edit global schedules. To define a definite run for a global schedule, navigate to **Administration > Schedules**.

To add a definite run to a custom schedule:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite, for which you want to add a definite run.
3. Click the **Schedule** tab.
4. Click the **Custom** option to enable the scheduling controls.
5. Click **Add Definite Run**.
6. On the **Configure Definite Run** page, click  and select the date and time when the execution plan, folder, or configuration suite, should definitely be run.
7. Click **OK**. Your definite run settings are listed on the **Schedule** page.
8. Click **Save** to add the definite run to the current schedule, or continue adding definite runs.

Editing Definite Runs

To edit a definite run:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite, for which you want to edit a previously configured definite run.
3. Click the **Schedule** tab.
4. In the **Actions** column of the definite run, click .
5. Edit the definite run criteria as required and then click **Save**.

Deleting Definite Runs

To delete a definite run:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan, folder, or configuration suite, for which you want to delete a previously configured definite run.
3. Click the **Schedule** tab.
4. In the **Actions** column of the definite run, click .

Schedule Page

Execution Planning > Details View > <Execution Element> > Schedule



Note: This page is not displayed for execution plans that are assigned to a testing cycle.

The **Schedule** page is used to define schedules for execution plans, folders, configuration suites, and report subscriptions.

The page provides the following schedule types:

| Schedule Type | Description |
|---------------|--|
| None | Click this option button so that the asset is not executed based on a schedule. All schedules included in a copy or baseline of a project are initially set to None . |
| Global | Click this option button to select a predefined schedule from the list box for the asset. Selecting a global schedule includes the schedule exclusions and definite runs which are defined in the global schedule. For information on defining global schedules, see the <i>Administration</i> topics in this Help. Selecting a global schedule displays the schedule details below the Custom option button. |
| Custom | Click this option button to define a custom schedule for the execution plan, folder, or configuration suite. Click Edit to edit the custom schedule in the fields below. |

For each selected schedule type, the page displays details. For a custom schedule, the details are editable.

 **Caution:** If tests assigned to an asset are not executed, the reason might be that the asset is still executing because of an earlier schedule, and the schedule interval is shorter than the duration of the execution. In this case, either increase the schedule interval, improve the performance of the executed tests, or remove tests from the execution plan. To view the application server logfile, click the **Application Server Log** tab in **Administration > Log Files**.

Schedule Details

The page displays details for the selected schedule type, which are editable for a custom schedule.

| Item | Description |
|--|---|
| From | Specifies when the execution schedule begins. Click  next to the specified date to change the date and time. |
| Interval | Specifies the interval at which the tests are executed. |
| Adjust schedule to daylight savings | Check this check box to automatically have your schedule adjust to daylight savings time.  Note: Daylight adjustment only works for intervals of two-hour multiples to avoid duplicate runs when setting time back one hour. |
| Run | Specifies when the execution ends: Forever Click this option button to specify that the execution should not stop executing. Time(s) Click this option button and select a number from the list box to define a specific number of executions. Until Click this option button to pick a specific time at which test executions are to end. Click  next to the specified date to change the date and time. |
| Exclusion | The exclusions that are defined for the schedule. To add an exclusion, click Add Exclusion . |
| Definite Runs | The definite runs that are defined for the schedule. To add a definite run, click Add Definite Run . Click  in the action column of a selected definite run to access the calendar tool and specify when the definite run is to take place. |

Executing Tests Assigned to Execution Plans

This section explains how to run tests that are assigned to execution plans with Silk Central.

You can run individual execution plans, folders, and configuration suites. In all cases you can define whether all assigned tests, or only tests matching certain status criteria should be executed. After an execution plan including automated tests is started, it is put in the central execution queue and an appropriate execution server is chosen to accomplish automated execution of the tests, based on the matching keywords and the server availability. For mobile device testing, a device is selected and reserved

for execution, if a mobile device selection is configured for this execution plan. If no execution server or no mobile device is available, the execution job stays in the queue until a suitable execution server and mobile device becomes available.

Starting Execution Plans

To run an execution plan independent of a schedule:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the project, execution plan, folder, or configuration suite that you want to execute, or use **Ctrl+Click** or **Shift+Click** to select multiple elements.
3. Click  (**Run**) on the toolbar. This is disabled for testing cycles, execution plans within testing cycles, and unassigned tests. The **Run** dialog box appears.
4. Define which tests you want to execute in the **Run** dialog box.
5. Check the **Go to Activities page** check box to advance to the **Activities** page after you define tests for execution.

Run Dialog Box

Execution Planning > Details View > <Execution Element> > Run

Execution Planning > Document View > <Execution Element> > Run

The **Run** dialog box enables you to specify which tests you want to execute based on filter criteria and to specify which product build the test should be run against. To open the **Run** dialog box, select an execution plan or an execution folder and click **Run** on the toolbar.

| Item | Description | | | | | | | | |
|--|--|--------|-------------|---------------------------|--|---------------------------------|--|-------------------------------|--|
| All tests | Select this option to execute all tests. | | | | | | | | |
| Tests... | Select this option to only execute tests that meet one of the following options: <table border="1" data-bbox="516 1144 1445 1449" style="margin-left: 20px;"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>With status Failed</td> <td>Check to re-execute all tests in the selected execution plan, that have the status <code>Failed</code>.</td> </tr> <tr> <td>With status Not Executed</td> <td>Check to re-execute all tests in the selected execution plan, that have the status <code>Not Executed</code>.</td> </tr> <tr> <td>Not run with build ...</td> <td>Check to re-execute all tests in the selected build that are not executed in the selected build. Tests that are executed in the selected or a later build are not re-executed.</td> </tr> </tbody> </table> | Option | Description | With status Failed | Check to re-execute all tests in the selected execution plan, that have the status <code>Failed</code> . | With status Not Executed | Check to re-execute all tests in the selected execution plan, that have the status <code>Not Executed</code> . | Not run with build ... | Check to re-execute all tests in the selected build that are not executed in the selected build. Tests that are executed in the selected or a later build are not re-executed. |
| Option | Description | | | | | | | | |
| With status Failed | Check to re-execute all tests in the selected execution plan, that have the status <code>Failed</code> . | | | | | | | | |
| With status Not Executed | Check to re-execute all tests in the selected execution plan, that have the status <code>Not Executed</code> . | | | | | | | | |
| Not run with build ... | Check to re-execute all tests in the selected build that are not executed in the selected build. Tests that are executed in the selected or a later build are not re-executed. | | | | | | | | |
| Tests that have had issues fixed since their last run | Select this option to only execute those tests that have had issues advanced to the <code>Fixed</code> state since the test's last execution. | | | | | | | | |
| Set build for execution plan | Select a past build from the Set build for execution plan list box to have the test run against a specific past build. This field defaults to the current build. Note that this option is not available if the execution plan is configured to read the build number from a build information file. If an execution folder contains execution plans with different product versions assigned to each, the build cannot be selected for the execution of the execution folder. | | | | | | | | |
| Run type | Choose Run as specified to run all selected tests with their own test type, or choose Run automated tests manually to re-run all selected tests manually. | | | | | | | | |

| Item | Description |
|------------------------------|---|
| Go to Activities page | Check this check box to advance to the Activities page after you define tests for execution. |

Manually Executing Automated Tests

Automated tests might sometimes deliver unexpected results because of environmental issues. When such a test fails in a system, and you want to verify if the failure is caused by a system component, or you know that the problem is caused by a currently dysfunctional system component, you can re-run the automated test manually.

To run an automated test manually, perform the following steps:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the project, execution plan, folder, or configuration suite that you want to execute, or use **Ctrl+Click** or **Shift+Click** to select multiple elements.
3. Click  (**Run**) on the toolbar. This is disabled for testing cycles, execution plans within testing cycles, and unassigned tests. The **Run** dialog box appears.
4. Define which tests you want to execute.
5. Select **Run automated tests manually** in the **Run type** list box. All selected tests are then treated as manual tests, and the **Current Run** page opens.

Managing Executions in External Execution Environments

With Silk Central 19.0 or later, Silk Central further extends its ability to integrate with existing tooling ecosystems by providing functionality to consume results for automated tests without having those automated tests executed through Silk Central execution servers.

This allows users with existing proprietary execution environments, which are too complicated or special to be adopted to the execution environment of Silk Central, or users with automated test results from various sources, for example continuous integration (CI) systems, to still collect all results of their whole testing process in one central place and to associate these results with their requirements, versions, and builds for holistic reporting and decision making.

The whole process around test creation, which means the representation of the actual test in Silk Central, and around execution planning does not change. You create tests in Silk Central and assign these tests to execution plans to execute them.

The difference is in the deployment settings of the execution plan. Instead of specifying an execution server, you enable the so called *External execution of assigned tests*.

Running such an execution plan then results in an execution plan run that is visible like normal execution plan runs in the Silk Central UI, with the difference that this execution plan is waiting to receive results from the external execution environment.

For external execution plan runs, you can create setup and cleanup tests in the same way as for runs that are executed on the Silk Central execution servers. For example, you can use a setup test to trigger the execution plan run in the external execution environment. This will block the Silk Central execution server for the duration of the external execution plan run, and then a cleanup test can be executed on the same execution server. Such setup and cleanup tests have the same ID as the external execution plan run. You can get the ID through the predefined parameter `#sctm_regular_execdef_run_id`. For additional information, see [Predefined Parameters](#).

If Silk Central is installed on your system, you can update such execution plan runs through the interactive REST API documentation, accessible from host:port/inst/Services1.0/swagger-ui.html, where host is your Silk Central host. For example <http://localhost:19120/Services1.0/swagger-ui.html>.

Using the REST API for External Executions

You can update execution plan runs that are executed from an external execution environment through the REST API, accessible from `host:port[/inst]/Services1.0/swagger-ui.html`, where `host` is your Silk Central host. For example `http://localhost:19120/Services1.0/swagger-ui.html`.

For authentication within the REST API calls, generate a web-service token in the **User Settings** page of the Silk Central UI. To access this page, hover the mouse cursor over the user name in the Silk Central menu and select **User Settings**.

You can also create a session ID for authentication within the API calls.

1. Click **Authorize** in the top right of the browser window. The **Available authorizations** dialog appears.
2. Enter your Silk Central credentials into the fields in the **Basic Authorization** area.
3. Click **Authorize**.
4. Close the **Available authorizations** dialog.
5. To login and create the session ID, expand the **login-api-controller** in the REST API.
6. Expand the **/login** API call.
7. Click **Try it out**.
8. Click **Execute**.
9. Copy the session ID from the Response body field to use it for further API calls.
10. Copy the session ID to use it for further API calls.

For information on how you can limit the usage of the REST API services for a web-service token or for a session ID, see *JMX Measures for Limiting Usage of REST Services*.

 **Note:** Micro Focus recommends authentication through a web-service token.

Filters

Filters provide an efficient means of finding exactly the information you need, while excluding extraneous detail. Filters highlight only those elements that are relevant to your needs, and enable you to quickly sort through requirements, test elements, and execution plans. By defining global filters, you can create complex filter criteria that are available throughout Silk Central without defining filter criteria each time you need to filter a list.

Based on your needs, you can create new filters, edit existing filters, select filters, delete filters, or turn filtering off at the project level. Projects do not contain default filters. You can access and edit filters from the toolbars in the Silk Central units and from the **Project Settings** unit.

 **Note:** Filters are not applied to reports. The **Recent Changes** filter enables you to view project-wide changes and additions that other users have made to tests since your last change acknowledgement. The **Show Changes/Show All** toggle button and the **Acknowledge** button in the **Tests** area help you to find out what changes other users have made. Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Creating Filters

To create a filter:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Filters** tab to view the list of available filters.
3. Click **New Filter**. The **New Filter** dialog box appears.
4. Type a **Name** for the new filter.

This name will be displayed in list boxes when the filter becomes available.

5. Select a **Category** from the list to make the filter available in the **Requirements, Tests, or Execution Planning** area of Silk Central.
6. *Optional:* Type a **Description** for the new filter.
7. *Optional:* Check the **Visible to other users** check box to allow other users to see the filter.
8. *Optional:* Check the **Editable by other users** check box to allow other users to edit the filter.



Note: If **Visible to other users** and **Editable by other users** are checked, the filter is public. To delete non-public (private) filters, you have to be the owner of the filter or you need the **Delete private filters of other users** permission.

9. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.



Note: You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

10. Select a **Property, Operator, and Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|--------------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

11. *Optional:* Click **More** if you want to add more than one filter category to the new filter. Repeat this procedure to define new categories.



Note: If you define more than one filter category, you must define whether the categories need to be fulfilled in addition to the existing categories (AND relationship), or if the filter returns true when one of the filter categories is fulfilled (OR relationship). Select either **AND** or **OR** to define the filter category relationship. You cannot define nested AND, OR relationships.

12. *Optional:* To remove filter categories, click **Fewer**. This removes the last filter category.
13. Click **OK** to save the new filter, or click **Cancel** to abort the operation.

Creating Advanced Filters

Advanced custom filters enable you to combine simple filters to create complex filters that apply multiple filter criteria simultaneously.

To create an advanced custom filter:

1. In the menu, select the appropriate area: **Requirements, Tests, or Execution Planning**.
2. Click  (**New Filter**) in the toolbar. The **New Filter** dialog box appears.

3. If necessary, click **Advanced** to show the whole dialog box.
4. Click **More** to display a second set of filter-parameter fields with which you can define a second set of filter parameters.
5. Select a logical operator for the application of the filtering queries. For example, with the operator `and` filtered elements must meet both sets of criteria and with the operator `or` filtered elements must meet one, but not both, of the criteria sets.
6. To delete a filter-parameter string, click .
7. To display additional filter-parameter fields and create additional filter queries, click **More**. To remove excess filter-parameter sets, click **Fewer**.

Editing Filters

To edit a filter:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Filters** tab to view the list of available filters.
3. Click the name of the filter you want to edit. The **Edit Filter** dialog box displays.
4. Edit the **Name** and **Description** of the filter.
5. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.



Note: You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

6. Select a **Property**, **Operator**, and **Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|--------------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

7. Click **OK** to save the edited filter definition.

Copying Filters

Create a new filter by copying an existing one and changing the settings.

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. In the toolbar of the selected area, select a filter from the list box.
3. Click . The **Edit Filter** dialog box opens with the settings of the original filter.

4. Edit the properties of the filter and click **OK**.

Deleting Filters

To delete a filter:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Filters** tab to view the list of available filters.
3. Click **Delete Filter** for the filter that you want to delete. A confirmation dialog box displays, asking you to confirm the deletion.
4. Click **Yes** to delete the selected filter or **No** to abort the operation. If you click **Yes**, you will be returned to the filters list and the deleted filter is no longer displayed.

Global Filter Details

Global filters allow you to quickly sort through elements in a Silk Central area, highlighting only those that are relevant to your needs.

To view the details of a global filter, click **Project:<Project Name> > Project Settings > Filters** in the menu, and click on the name of the filter.

The specific Silk Central area in which a new filter is available is defined by selecting the general filter category.

Each global filter has to have a property, an operator, and a value from the respective list boxes:

- Property** Available properties depend on the filter category. It defines the property for which you are defining a filter setting. If you selected an attribute category, the property list includes custom attributes to query against.
- Operator** Specifies the filter operator. The operator depends on the property type. For example, if you select a property that is based on a string field type, the available operators are = (equals defined value), *not* (differs from the defined value), *contains* (contains the defined value somewhere in the string), and *does not contain* (does not contain the defined value in the string).
- Value** The value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box.

Recent Changes

The *Recent Changes* filter enables you to efficiently view and acknowledge changes and additions that other users have made to requirements, tests, or execution plans project-wide since your last change acknowledgement.

The two buttons at the far-right of the toolbar, the **Show Changes/Show All** toggle button and the **Acknowledge** button, help you to find out what changes other users have made.



Note: Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Applying Filters

After you have created and stored a custom filter, you can apply that filter to the selected tree. Custom filters can be applied for requirements, tests and execution plans. Only elements that meet applied filter criteria are displayed in the tree.



Note: Filtered requirements are returned in read-only form and can not be edited. The **Edit Properties** button is disabled for filtered requirements.

To apply a stored filter:

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. Select the desired filter from the **Filter** list box on the toolbar.
3. All elements that meet the filter's criteria are displayed.



Note: To remove filtering and display all elements, select **<No Filter>** from the **Filter** list box on the toolbar.

Screen Capturing

You can enable screen capturing for a test element or an execution plan. The screenshots are added to the result files in the **Files** page and the **Timeline** page of the **Test Run Results** dialog box. You can then analyze the screenshots, for example if errors have occurred during tests.

The **Screen Capturing** setting in the **Deployment** page of an execution plan can override the setting for a specific test element, which is included in the execution plan.

Enabling Screen Capturing

To enable screen capturing for a test:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select a container, folder, or test.
3. Click the **Parameters** tab.
4. Click **Add Custom Parameter**. The **Set Custom Parameter** dialog box opens.
5. In the **Name** field, type `captureScreen`.
6. From the **Type** list box, select `String`.
7. In the **Value** field, type one of the following:
 - `onError`, if you want to add a screenshot to the result files when an error occurs during the execution of the execution plan.
 - Any other string or nothing, if you do not want to add a screenshot to the result files.

Video Capturing

You can enable video capturing for a test element or an execution plan. The videos are added as a WMV video file to the result files in the **Files** page of the **Test Run Results** dialog box. You can then analyze the videos, for example if errors have occurred during tests.

The **Video Recording** setting in the **Deployment** page of an execution plan can override the setting for a specific test element, which is included in the execution plan.

Limitations:

- Video recording is only supported on Microsoft Windows client machines.
- Video capturing on Microsoft Windows 10 N and KN editions is not supported.
- Video recording of WPF windows is not supported on Windows 7 Classic Theme.
- Recording videos at 4k resolution is not supported.



Note: To use video capturing on a Windows Server 2008 execution server, you need to install the following server features:

- Desktop Experience
- Quality Windows Audio Video Experience

To use video capturing on a Windows Server 2012 execution server, you need to install the following server features:

- Desktop Experience
- Media Foundation



Attention: The process of capturing videos is very CPU-intensive. Use this functionality only when needed. To use this functionality you require a system with sufficient CPU power, for example a system with multiple CPUs.

Capturing a video for each individual test case in a suite is supported out of the box for JUnit 4 tests and Silk Test Classic Plan tests. To use this feature for other test types, use the methods in the `TestCaseStartFinishSocketClient` interface. For additional information, see *Indicating Start and Finish for Video Capturing*.

Enabling Video Capturing

To enable video capturing for a test:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select a container, folder, or test.
3. Click the **Parameters** tab.
4. Click **Add Custom Parameter**. The **Set Custom Parameter** dialog box opens.
5. In the **Name** field, type `recordVideo`.
6. From the **Type** list box, select `String`.
7. In the **Value** field, type one of the following:
 - `always`, if you want to add a video to the result files on every execution of the execution plan.
 - `onError`, if you want to add a video to the result files when an error occurs during the execution of the execution plan.
 - Any other string or nothing, if you do not want to add a video to the result files.

Automatically Deleting Results and Result Files

Define rules to clean up old results and result files of execution plan runs to reduce database size and to improve the manageability of your execution plan runs.

You need the **Delete execution plan runs** permission to perform this task.

The amount of collected results and result files increases over time as you run your execution plans. After a while, some results may not be relevant anymore, and managing your results is easier if only the important information is retained. In Silk Central you can define rules for the automatic cleanup of results and result files that you no longer require.

1. In the menu, click **Execution Planning > Details View**.
2. Select the node in the **Execution Plans** tree for which you want to configure result cleanup rules. All child nodes inherit the defined rules. To define cleanup rules specifically for unassigned tests, select the **Unassigned Tests** node in the **Execution Plans** tree. Unassigned tests include items such as uploaded Silk Performer tests and try runs.
3. In the **Result Cleanup Rules** section on the **Properties** page, click **New Rule**. The **New Result Cleanup Rule** dialog box appears.
4. Specify the **Rule type**. You can choose to delete only result files, or entire runs, which includes results too.
5. In the **Delete after** list, select the time after which the rule triggers the deletion. Cleanup rules only trigger when new results are stored for the selected node.



Note: Runs and result files are not deleted if one of the following applies:

- A test run with a tagged build is contained
- A test run is contained, which is the latest run of this test in this execution plan
- An unassigned test is contained, which is the latest run of this test
- There is only one execution plan run

6. If you have selected the rule type **Delete result files**, you can limit the deletion to specific file types in the **Only delete files of type** field. Enter the file type extensions as comma-separated list, for example `bmp, png`. Leave this field empty to delete all result files.

7. Click **OK**.

To update or remove a cleanup rule, click the **Delete** or **Edit** icon. You can also deactivate a cleanup rule by toggling the **Status**. Inherited rules can only be modified on the top node on which they were created.

Execution Planning Views

Silk Central offers several views and tools to plan and execute your tests.

Working with the Execution Plans Tree

Execution Planning > Details View

Folders, execution plans, configuration suites, and testing cycles are displayed, organized, and maintained through a hierarchical tree structure, the **Execution Plans** tree. You can use any number of hierarchy levels within the tree.



Note: When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

Keyboard Functions (Execution Planning)

The following actions can be performed on the selected nodes:

| Keys | Action |
|--------------------|--|
| Up | Move selection up |
| Down | Move selection down |
| Right | Expand element |
| Left | Collapse element |
| F2 | Edit element |
| Ctrl+C | Copy element |
| Ctrl+X | Cut element |
| Ctrl+V | Paste element |
| Del | Delete element |
| Click | Select an element and remember it as the current element. |
| Ctrl+Click | Toggle the selection status of the clicked element and add/remove additional elements with further clicks. |
| Shift+Click | Select the span from the currently-selected element to a newly selected element. |
| Shift+Up | Select element above. |

| Keys | Action |
|-------------------|--|
| Shift+Down | Select element below. |
| Enter | Assign element (only in assignment tree) |

Expanding or Collapsing the Execution Plans Tree

You can consolidate levels of the **Execution Plans** tree or display all levels of the hierarchy based on your viewing needs.

To collapse or expand levels of the **Execution Plans** tree:

1. In the menu, click **Execution Planning > Details View**.
2. Select a folder in the **Execution Plans** tree and continue with one of the following options:
 - Click  to the left of the name of the folder to expand the folder.
 - Click  to the left of the name of the folder to collapse the folder.
 - Right-click on the folder and select **Expand** or **Collapse**.

Editing Elements of the Execution Plans Tree

To edit an existing execution plan element:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan, folder, testing cycle, or configuration suite you want to edit.
3. In the **Details** section, click **Edit**. The **Edit** dialog box appears.
4. Edit the element by modifying the criteria in the dialog box.

For execution plans, if there are no runs and no tests assigned to the execution plan, you can choose an alternative test container for the execution plan from the **Test Container** list box.
5. Click **OK** to save the edited element.

Copying Elements of the Execution Plans Tree

To copy and paste an execution plan, a folder, or a configuration suite:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the element that you want to copy.
3. Click  on the toolbar or right-click the element and select **Copy**.
4. Select the target folder where the element is to be pasted.
5. Click  on the toolbar or right-click the target node and select **Paste**. The **Execution Plans** tree is updated with a copy of the pasted element. All assigned tests, filters, and scheduling parameters are copied along with the element.

Deleting Elements of the Execution Plans Tree

To delete an execution plan, a folder, or a configuration suite:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the element that you want to delete.
3. Click  on the toolbar or right-click the element node and select **Delete**.
4. Click **Yes** on the deletion confirmation dialog box to remove the element from the **Execution Plans** tree.

When deleting an element of the **Execution Plans** tree, the run results of the assigned tests are also deleted. The test run results may still appear in reports, because they are stored in the database, which is not immediately updated after the deletion of the element.

Filtering the Execution Plans Tree

Follow the steps below to filter the **Execution Plans** tree so that only the selected node and child nodes are visible. This is useful if you want to print a sub-set of execution plans to a PDF.

1. Create a custom filter that defines the execution plans that you want to view.
2. Select a folder or configuration suite in the tree view, right click and select **Filter Subtree**.

Printing Assigned Tests

For each execution plan in the **Execution Plans** tree, you can print a document that includes information about all tests that are assigned to the execution plan. You can print this document to PDF or a local printer.

To print information about all tests that are assigned to an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, right click the execution plan and click **Print Assigned Test Details**. The **Test Details** dialog box appears.
3. To print the assigned test information to a local printer, click **Print**. To save the information as a PDF document, click **Download as PDF**.

Testing Cycle Properties Page

Execution Planning > Details View > <Testing Cycle> > Properties

The **Testing Cycles Properties** page is available when you select any testing cycle from the **Execution Plans** tree and then click **Properties**.

Click **Edit Properties** to display the **Edit Testing Cycle** dialog box where you can edit the **Name**, **Description**, and **Planned Time** of the testing cycle.

The following table displays the available testing cycle properties.

| Property | Description |
|---------------------------------|--|
| Testing Cycle Name | Name of the testing cycle. Edit this value in the Edit Testing Cycle dialog box. Click  to locate the testing cycle in the Manual Execution Planning unit. |
| Testing Cycle ID | Unique ID of the testing cycle. |
| Description | Description of the testing cycle. Edit this value in the Edit Testing Cycle dialog box. |
| Status of Last Run | Status is an aggregate of the last run of all test statuses within the execution plans of the testing cycle: Passed , Failed , Not Executed , or N/A . |
| Start Time of Last Run | Start time of the last run. |
| Testing Cycle Start Date | The scheduled start date for the testing cycle. |
| Testing Cycle End Date | The scheduled end date for the testing cycle. |
| Capacity [hh:mm] | Amount of time that you expect the testing cycle to take. Edit this value in the Edit Testing Cycle dialog box. If you assign manual testers to the testing cycle, the value for Capacity is the sum of the Capacity for each manual tester. |

| Property | Description |
|-----------------------------|---|
| Planned Time [hh:mm] | Cumulative planned time of all manual tests within all assigned execution plans of the testing cycle. |
| Time Left [hh:mm] | This is the remaining time available after subtracting Planned Time from Capacity . |
| Created On | Date that the testing cycle was created. |
| Created By | User that created the testing cycle. |
| Changed On | Date that the testing cycle was last modified. |
| Changed By | User that last modified the testing cycle. |

Assigned Manual Testers

This section lists all of the manual testers included in the cycle.

| Property | Description |
|---------------------------|---|
| Tester | Name of the tester. |
| Capacity | Enter the amount of time that this tester is available for the testing cycle in hh:mm format. The sum of all capacity values for each tester will be the testing cycle Capacity value. |
| Add/Remove Testers | Click to display the Manual Testers dialog box which you use to add testers to the testing cycle. |

Execution Planning Toolbar Functions

Execution Planning > Details View

Execution Planning > Document View

The executions toolbar provides important commands for managing your execution plans.

 **Note:** Some commands in the execution planning toolbar are also available through context menus in the **Execution Plans** tree.

The following commands are included in the execution planning toolbar:

| Command | Icon | Description |
|--------------------------------------|---|--|
| Details View |  | Displays the Details View , which enables you to drill deeply into the properties of a single element of the Execution Plans tree. |
| Document View |  | Displays the Document View , which shows selected properties of all executions in a single view. |
| New Child Folder |  | Add a new folder to the selected project or folder. This command is disabled for execution plans that are assigned to a testing cycle. |
| New Child Testing Cycle |  | Add a new testing cycle to the selected project or folder. |
| New Child Configuration Suite |  | Add a new configuration suite to the selected project or folder. This command is disabled for execution plans that are assigned to a testing cycle. |
| New Child Execution Plan |  | Add a new execution plan to the selected project, folder, or configuration suite. This command is disabled for execution plans that are assigned to a testing cycle. |
| Edit |  | Open the selected element of the Execution Plans tree for editing. |

| Command | Icon | Description |
|------------------|---|---|
| Delete |  | Deletes the selected element of the Execution Plans tree. This command is disabled for execution plans that are assigned to a testing cycle. |
| Cut |  | Cut an element of the Execution Plans tree and save it to the clipboard. This command is disabled for execution plans that are assigned to a testing cycle. |
| Copy |  | Copy an element of the Execution Plans tree to the clipboard. This command is disabled for execution plans that are assigned to a testing cycle. |
| Paste |  | Paste an element from the clipboard to the Execution Plans tree. This command is disabled for execution plans that are assigned to a testing cycle. |
| Paste as Child |  | Paste an element from the clipboard as a child to the currently selected element of the Execution Plans tree. This command is disabled for execution plans that are assigned to a testing cycle. |
| Run |  | Start the execution of the tests that are assigned to the execution plan. This command is disabled for execution plans that are assigned to a testing cycle. |
| New Filter |  | Create a new custom filter for the Execution Plans tree. |
| Filters | | Lists the available filters for the Execution Plans tree |
| Edit Filter |  | Edit the currently selected custom filter. |
| Delete Filter |  | Delete the currently selected custom filter. |
| Copy Filter |  | Copy the currently selected custom filter. |
| Download as PDF |  | This function is enabled only in the Document View . The action generates a PDF that includes all currently visible nodes of the Execution Plans tree. |
| Download as XLSX |  | This function is enabled in the Document View . It generates an Excel sheet that includes all currently visible rows. |

Execution Plan Properties Page

Execution Planning > Details View > <Execution Tree Element> > Properties

The **Properties** page offers detail on all properties and relevant information for the selected project, folder, execution plan, testing cycle, or configuration suite. You have the option of expanding and collapsing the information in each section on this page. This enables you to customize the information you see to your specific needs.

Details

The **Details** section displays basic information about the selected element. For additional information, see [Creating Execution Plans](#) and [Editing Elements of the Execution Plans Tree](#).

Execution Status

The **Execution Status** section displays information about the latest runs. The **Status of last run** displays the cumulated test status of the latest run of the execution plan, or the cumulated test status of all contained execution plans, for a folder or configuration suite. The **Duration** displays the duration of the

latest run of an execution plan, or in case of a folder or configuration suite, the cumulated duration of the latest runs of all contained execution plans. In the simplest case, for automated tests on a single execution server or only manual tests, the duration is the time displayed for the latest run on the **Runs** page. If the last execution involved both automated and manual tests, only the automated or manual tests will be considered, depending on which test was executed later. If the execution plan contains multiple tests, the duration is measured from the time when the first test begins executing and ends when the last test completes execution. This includes the overhead time, which is needed for stopping/starting tests between executions.

For additional information, see [Analyzing Test Runs](#).

Test Execution Setup

The **Test Execution Setup** section displays information about assigned tests or testers. For additional information, see [Managing Execution Plans Folders and Configuration Suites](#) and [Test Assignment](#).

Attributes

Attributes are freely definable characteristics that can be applied to execution plans. For additional information, see [Execution Plan Attributes](#).

Parameters

The **Parameters** section displays information about parameters, which are freely configurable input values to execution plans. For additional information, see [Execution Plan Parameters](#).

Notifications and Reports

The **Notifications and Reports** section includes check boxes which allow you to check whether you want to be notified based on the outcome of the execution. Notification only works if an email server has been configured by your administrator. You also have to specify an email address for your account in **<User Name> > User Settings** . Additionally, you can trigger report subscriptions based on the outcome of the execution. The list of available reports is based on the reports that have a subscription defined. For additional information, see [Managing Report Subscriptions](#).

| Check Box | Description |
|---|---|
| Execution finishing successfully | Check to receive a notification email each time an execution run finishes successfully. |
| Execution finishing with not passed tests | Check to receive a notification email each time an execution finishes with status <code>not executed</code> or <code>failed</code> . |
| Execution finishing with changed number of not passed tests | Check to receive a notification email each time the number of failed or not executed tests changes in comparison to the previous run, when an execution finishes. |

Result Cleanup Rules

Use the **Result Cleanup Rules** section to define rules to clean up old results and result files of execution plan runs. For additional information, see [Automatically Deleting Results and Result Files](#).

Execution Planning Document View

Execution Planning > Document View

The **Execution Planning** area offers two views of execution properties: **Document View** and **Details View**. **Document View** provides a high-level, read-only view of the defined properties and characteristics

of the latest run of all execution plans, folders, and configuration suites in the selected project. Use the **Document View** to plan the execution of your execution plans. You can use filters to obtain an execution-duration estimate of the filtered execution-plan subset.

The **Document View** displays the following columns for a selected element in the **Execution Plans** tree:

| Column | Description |
|-----------------------|--|
| Status | The cumulated test statuses of the latest run of the execution plan, or the cumulated test statuses of all contained execution plans, for a folder or configuration suite. If you apply a filter to the Execution Plans tree, the test status totals are based on the filtered execution plans only. |
| Assigned Tests | Total number of assigned tests. If you apply a filter to the Execution Plans tree, the amount of the assigned tests is based on the filtered execution plans only. |
| Build | Build that is used when the execution plan is executed. |
| Version | Version that is used when the execution plan is executed. |
| Product | Product that the associated test container is based on. |
| Priority | For automated execution plans, when more than one execution plan is queued but only one execution server is available, the Priority determines which execution plan is executed first. The same is true if execution servers are available but multiple execution plans require an unavailable mobile device. |
| Last Execution | Last time the execution plan was started. |
| Planned Time | Total planned time for all manual tests. If you apply a filter to the Execution Plans tree, the planned time is based on the filtered execution plans only. |
| Duration | Duration of the latest run of an execution plan, or in case of a folder or configuration suite, the cumulated durations of the latest runs of all contained execution plans. If you apply a filter to the Execution Plans tree, the duration is based on the filtered execution plans only. |
| Next Execution | Next scheduled execution. |
| Test Container | Test container containing the tests that can be assigned to this execution plan. |
| Attributes | If you have assigned attributes to your execution plans, you can select to display them in the grid. Attributes are freely definable characteristics that can be applied to execution plans. You can use attributes for filtering and reporting purposes. |



Note: Execution plan totals accumulate to the parent level, for all currently displayed execution plans. If you apply a filter to the **Execution Plans** tree, the execution plan totals are based on the filtered execution plans only. For example, the execution plan totals include execution runs from all child execution plans that are selected with the filter, and the project totals include execution runs from all execution plans that are selected with the filter.

Current Run Page

Execution Planning > Details View > <Execution Plan> > Current Run

To access the **Current Run** page of an execution plan, select the execution plan in the **Execution Plans** tree and click the **Current Run** tab. The **Current Run** page displays all information related to the active test runs of the selected execution plan.

Manual Test Runs or Business Process Test Runs

The **Current Run** page displays the active manual test run or business process test run, until the test run is finished. For manual tests that are assigned to a testing cycle, this page is renamed to **Run** page. In such a case, the **Run** page continues to display the run, after the run is finished. The page features two grid views, **Assigned Tests** and **Test Steps**. **Assigned Tests** shows information about the active manual test run or business process test run, and **Test Steps** shows information for each of the steps in the manual test or business component. You can filter the test runs in the **Assigned Tests** view by selected columns. The **Test Details** and **Step Details** views show additional information.

Click **Reload** to refresh the **Current Run** page. If you have changed manual tests or business process tests, click **Synchronize Current Runs**. This will update all assigned tests of the current runs.

The following information can be synchronized when you click **Synchronize Current Runs**:

- Changes to the test assignment of an execution plan.
- Updates to test properties in the **Tests** area.
- Updates to test step properties, both for manual tests and business process tests.
- Updates to assigned unfinished business components in business process tests.

You can select which information to synchronize in the **Synchronize Current Runs** dialog.



Note: For business process tests, changes to the component assignment and to finished components are not synchronized.

If other execution plan runs are started while the **Current Run** has not been finished, a note displays, stating that newer runs are available. You can see information on those runs on the **Activities** page.

Automated Test Runs

For automated tests, the **Current Run** page shows the progress of the execution. Click **Update** to refresh the **Current Run** page. For automated tests that run on a mobile device, click **Show Screencast** to view a live stream of the activities on the device. Live streams are not available for Sauce Labs devices.

Execution Planning History Page

Execution Planning > **Details View** > <**Project**> > **History**

The **History** page details the execution planning history of the selected project. You can display or hide columns, adjust the width, and drag the columns around to change their order. Click on the columns to filter the content in the grid. The following actions generate execution planning history entries:

- Deleting an execution plan, folder, testing cycle, configuration suite, or a configuration



Note: When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

Working with Silk Performer Projects

Silk Performer is fully integrated with the test and execution functionality of Silk Central. Silk Performer projects can be integrated into Silk Central tests and directly executed through Silk Central. This allows for powerful test-result analysis and reporting. It also enables unattended testing, which means tests are run automatically by Silk Central based on pre-configured schedules.

Refer to the Silk Performer Help for details on configuring the integration of Silk Performer with Silk Central.

Silk Performer project files can be directly opened in Silk Performer from Silk Central, where scripts and settings can be edited. Edited Silk Performer projects can subsequently be checked back into Silk Central to make them available for future test executions.



Note: The term Project is used differently in Silk Performer than it is in Silk Central. A Silk Performer project, when uploaded to Silk Central, becomes the core element of a Silk Central test. Silk Central projects are high-level entities that may include multiple Silk Performer projects, tests, execution plans, and requirements.

Downloading Silk Performer Projects

Whereas opening a Silk Performer project may involve checking out a Silk Performer project from a source-control tool, editing the project in Silk Performer, and checking the project back into Silk Central, *downloading* a project involves downloading a copy of a project and working with it independently of Silk Central. Changes you make to downloaded projects are not automatically migrated back to Silk Central.

To download a Silk Performer project:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select a Silk Performer test.
3. Click the **Properties** tab.
4. Scroll down to the **Silk Performer Test Properties** section.
5. Click . A file download dialog box displays, asking you to confirm that you wish to download the specified Silk Performer project to your local system.
6. Click **Save** to open the file in Silk Performer. If not already open in the background, Silk Performer is invoked.
7. The **Select Target Directory** dialog box displays, loaded with the default directory path to which the specified Silk Performer project will be saved. If you approve of the specified pathname, click **OK**, otherwise click **Browse** to specify an alternate path.



Note: Even if you have configured source-control integration, you will not be prompted to check out the Silk Performer project from your source-control system because you are working with this file independently of Silk Central.



Note: Silk Performer projects utilized by Silk Central can also be downloaded directly from the Silk Performer user interface. For additional information, refer to the Silk Performer documentation.

Opening Silk Performer Projects

To open a Silk Performer project from Silk Central:

1. In the menu, click **Tests > Details View**.
2. In the **Tests** tree, select a Silk Performer test.
3. Click the **Properties** tab.
4. Scroll down to the **Silk Performer Test Properties** section.
5. Click . A file download dialog box displays, asking you to confirm that you wish to open the specified Silk Performer command file (.spwbcmd) in Silk Performer.
6. Click **Open** to open the file in Silk Performer. If not already open in the background, Silk Performer is invoked. The **Select Target Directory** dialog box opens, loaded with the default directory path to which the specified Silk Performer project will be saved.
7. If you approve of the specified pathname, click **OK**, otherwise click **Browse** to specify an alternate path.
8. If you have configured source control integration for Silk Central, for example StarTeam, you are presented with a login screen for your source control client. Enter valid user connection settings and click **OK** to continue.



Note: Silk Performer projects utilized by Silk Central can also be opened directly from Silk Performer. For additional information, refer to the Silk Performer documentation.

Accessing Execution Plan Run Properties

Silk Central provides information on execution plan run properties during Silk Performer test executions. Use the `AttributeGet` methods to access execution plan run properties in the Silk Performer script. You can access the following properties in the script:

| Parameter | Description |
|---|--|
| <code>#sctm_execdef_name</code> | The name of the execution plan that is currently executed. If the execution was triggered from the Tests area, the parameter has the value <code>Unassigned Tests</code> . |
| <code>#sctm_execdef_id</code> | The numeric identifier (database key) of the execution plan that is currently executed. |
| <code>#sctm_product</code> | The name of the product as defined in the executed test container. |
| <code>#sctm_version</code> | The name of the version to which the results of the execution are associated. |
| <code>#sctm_build</code> | The name of the version to which the results of the execution are associated. |
| <code>#sctm_keywords</code> | A comma separated list which contains the keywords that are defined for this execution plan. |
| <code>#sctm_regular_execdef_run_id</code> | The numeric identifier (database key) of the regular execution plan run. For setup and cleanup tests this property contains the run ID of the regular run. |
| <code>#sctm_test_results_dir</code> | The path of the directory where the result files of the test reside. |
| <code>#sctm_test_name</code> | The name of the Silk Central test being executed. |
| <code>#sctm_test_id</code> | The numeric identifier (database key) of the test being executed. |
| <code>#sctm_source_root_dir</code> | The local path of the test container's root directory, where all files under version-control reside. |
| <code>#external_id</code> | The unique identifier of a test case in a test package to be executed. You can use this as a filter if only specific test cases in a test package should be executed. |
| <code>#is_test_package_root</code> | If <code>true</code> , the test being executed is the root node of the test package (default). If <code>false</code> , only specific test cases in a test package specified by <code>#external_id</code> are executed. |
| <code>#sctm_data_driven_parent_test_name</code> | The name of the parent test, if the executed test is a data driven row. |
| <code>#sctm_data_driven_parent_test_id</code> | The numeric identifier (database key) of the parent test, if the executed test is a data driven row. |

Executing Attended Silk Performer Tests

Attended tests are Silk Performer tests that are executed manually in Silk Performer and are not executed automatically based on a predefined schedule in Silk Central.

 **Note:** To use Silk Central's data-driven test functionality with Silk Performer scripts, data sources with column names matching the corresponding Silk Performer project attributes must be used in conjunction with `AttributeGet` methods (password attributes cannot be passed and will always use the values that are specified in the script).

To execute an attended test run in Silk Performer:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a Silk Performer test.
3. Click the **Properties** tab.
4. Scroll down to the **Silk Performer Test Properties** section.
5. Click . A file download dialog box displays, asking you to confirm that you wish to run the specified Silk Performer command file (.spwbcmd).
6. Click **Open** to open the project in Silk Performer. If not already open in the background, Silk Performer is invoked. The **Select Target Directory** dialog box displays, loaded with the default directory path to which the specified Silk Performer project will be saved.
7. If you approve of the specified pathname, click **OK**, otherwise click **Browse** to specify an alternate path. The **Silk Performer Workload Configuration** dialog box opens with all of the workload settings that are associated with the Silk Performer project.
8. Edit the workload settings as required and click **Run** to begin the test and monitor the test results with Silk Performer.

 **Note:** Clicking **Run** without editing any workload settings executes the Silk Performer test in exactly the same way as if the test had been executed directly from Silk Central as an unattended test.

Editing Silk Performer Test Properties

 **Note:** To use the data-driven test functionality of Silk Central with Silk Performer scripts, you have to use data sources with column names that match the corresponding Silk Performer project attributes in conjunction with `AttributeGet` methods.

To edit Silk Performer test properties:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a Silk Performer test.
3. Click the **Properties** tab.
4. Scroll down to the **Silk Performer Test Properties** section.
5. Click **Edit Silk Performer Test Properties**.
6. Proceed with the configuration of your Silk Performer test.

Analyzing Silk Performer Test Results

Performance Explorer enables in-depth analysis of Silk Performer test results. The **Analyze Results** option downloads only selected results, in contrast to “Downloading result packages”. To assist you in analyzing the results of your optimization efforts, Performance Explorer even allows you to compare statistics from multiple test runs side-by-side in custom comparison reports.

The results of tests that are run using Silk Central can be automatically loaded into Performance Explorer through commands on the **Runs** page in the **Tests** area.

For full details on using Performance Explorer and integrating Silk Performer with Silk Central, refer to the Silk Performer documentation.

To open Silk Performer test results in Performance Explorer:

1. In the menu, click **Tests > Details View** .
2. Select the test you are interested in viewing.
3. Click the **Runs** tab.
4. Click  in the **Actions** column of the test execution for which you want to download results. A **File Download** dialog box displays, showing you the name of the Performance Explorer command file, `.spppecmd`, that you are about to download.
5. Click **Open** to open the results in Performance Explorer. Alternatively, you can click **Save** to save the results locally.
6. If not already open in the background, Performance Explorer now opens, connected directly to your Silk Central installation, and fetches the results of the selected execution run.



Note: To prepare for a custom comparison report that compares the results of multiple executions in a single report, you may download the results of additional executions from the **Runs** page. Additional execution results are displayed in the existing instance of Performance Explorer on the Silk Central ribbon. For additional details on custom comparison reports, refer to the Performance Explorer documentation.

Downloading Silk Performer Test Result Packages

Downloading result packages is the ideal option if you want to analyze the complete results set of a test run, or if you want to download the complete results set for offline analysis. Because result packages often include large TrueLog On Error files, result packages can be compressed and downloaded to your local hard drive as `.lrx` files. Downloading results locally can also be useful when you are working from a slow Internet connection.

To download Silk Performer test results:

1. In the menu, click **Tests > Details View** .
2. In the **Tests** tree, select a Silk Performer test.
3. Click the **Runs** tab.
4. Click  in the **Actions** column of the test execution for which you want to download results. A **File Download** dialog box displays, showing you the name of the compressed results package file, `.lrx` that you are about to download.
5. Click **Open** to open the results in Performance Explorer. Alternatively, you can click **Save** to save the results locally.
6. If not already open in the background, Performance Explorer now opens. You are presented with an **Import Project** dialog box that indicates the target directory to which the results will be saved. Click **OK** to accept the default path, or click **Browse** to select an alternate path. The downloaded results are then displayed in Performance Explorer.



Note: If you accept the default projects directory where result packages are typically stored, which we generally recommend, then the results will be stored with all other Silk Performer results and will be readily accessible through the Performance Explorer `Add Loadtest Results` command.

Uploading Silk Performer Test Results

Once you have completed running an attended test in Silk Performer, you can upload the test results to Silk Central and associate the results with a test.

To upload results from an attended Silk Performer test:

1. Run an attended Silk Performer test.
For additional information, see *Executing Attended Silk Performer Tests*.
2. When the test is complete, select **Upload Results to Silk Central** from the **Results** menu.
The **Login** screen of the **Upload Results to Silk Central** wizard displays.

3. Enter your **Password** and click **Next**.



Note: Because this is an attended test, the wizard already knows the appropriate hostname and username of the test to which these results are to be uploaded.

4. If not already selected by default in the project list, select the Silk Central project to which you want to upload the Silk Performer results.

5. If not already selected by default in the tree list, select the test to which you want to upload the results. Click **Next**.



Note: You can right-click in the tree and use the commands on the context menu to create a new test, child test, test folder, or child test folder to which the results can be saved.

6. On the subsequent screen you can specify Version and Build numbers for the assigned Product to which the uploaded results belong. Also specify the Silk Performer test result status, for example `Passed`, `Failed`.



Note: If any errors occurred during the test run, the test result status is set to `Failed` by default.

7. Click **Finish** to upload the results. The uploaded results are displayed in **Tests > Details View > <Silk Performer Test> > Runs**.

Working with Silk Performance Explorer

Silk Performance Explorer (Performance Explorer) is used for in-depth analysis of test runs. Performance Explorer results analysis can be started directly from the **Execution** area and the **Tests** area of Silk Central through execution runs on the **Runs** page or from Performance Explorer itself. Refer to the Performance Explorer documentation for details regarding the integration of Performance Explorer with Silk Central.

The results of load-test runs in Silk Performer can also be uploaded to Silk Central and associated with tests. Refer to the *Silk Performer Help* for more details.

For additional information about the integration of Silk Central integration with Silk Performer, refer to the *Silk Performer Help* and the *Performance Explorer User Guide*.

Silk Test Classic Tests

This section describes how to execute tests in Silk Test Classic.

Adding a Silk Test Classic AUT Host

For execution plans that run Silk Test Classic tests, you may have a setup where the Silk Test Classic agent is on a different computer than the execution server. In this case, you can define the location of the Silk Test Classic agent, the Silk Test Classic AUT (Agent Under Test) Hostname.

To add a a Silk Test Classic AUT host to the selected execution plan or edit the host:

1. In the menu, click **Execution Planning > Details View**.
2. Select the execution plan to which you want to assign the Silk Test Classic AUT host.
3. Click the **Deployment** tab.
4. In the **Silk Test Classic AUT Hostname** section, click **Edit**. The **Edit Silk Test Classic AUT Hostname** dialog box displays.
5. In the **Hostname** field, type the name of the computer where the Silk Test Classic agent runs. Proper configuration of option files is required. For details on the command-line option `-m`, refer to the Silk Test Classic documentation.
6. Click **OK** to add the Silk Test Classic AUT host to the selected execution plan.

Automated Execution of Silk Test Classic Tests

All tests within an execution plan use the same Silk Test Classic instance for tests. The Silk Test Classic GUI is opened once with the first Silk Test Classic test execution and is closed automatically after the last Silk Test Classic test execution. Each Silk Test Classic test execution produces its own results. If for any reason the Silk Test Classic GUI closes during a test, it will reopen automatically with the next Silk Test Classic test execution.

Automated Execution of Data-Driven Silk Test Classic Test Cases

If the **data driven** check box is checked in Silk Test Classic test properties, each Silk Test Classic test will be repeated once for each data row in the external data source. By default, plan-based execution mode is used for data driven tests. This means that the results of all data rows will be listed under a single node in the `.res` result file. If execution mode is switched to script-based data driven in `ScExecServerBootConf.xml`, a result node will be created in the `.res` result file for each data row.

Specifying Agent Under Test (AUT)

When a Silk Test Classic agent cannot run on the same machine as the Silk Central execution server, for example when tests are run on platforms other than Windows, the hostname and port may be specified by the Silk Test Classic AUT Hostname setting in the **Deployment** page of an execution plan. If the setting has not been defined, Silk Test Classic default values are used, for example from `partner.ini`. The syntax for AUT is `hostname:port`. The agent must be started manually prior to test execution and configured to listen at the specified port. By default, the TCP/IP protocol is used for communication between Silk Test Classic instances and Silk Test Classic agents. Ensure that both programs have been configured to use the same protocol.



Note: Be careful when you have multiple execution servers assigned to an execution plan as Silk Test Classic agents can only work with one Silk Test Classic instance at a time.

Silk Test Classic Time-out Settings

If you have Silk Test Classic test cases that require more than 1 hour to complete, you must adjust the time-out settings in Silk Central. Otherwise, Silk Central assumes that something has gone wrong in the execution and terminates Silk Test Classic. For details about setting the Silk Test Classic time-out, see the *Administration* topics in this Help.

Silk Test Classic Logs

The RMS log file in Silk Test Classic is used to log data for each test case as test runs progress. Three types of data records are written to this file: status, memory and user records. By monitoring this file, the RMS Remote Agent has a means of determining the progress of each test run.

You can write your own comments into the user records of the log file by executing the `PrintToRMSLog 4Test` function.

Examples:

```
PrintToRMSLog ("Error", "An intended error")
PrintToRMSLog ("Info", "testcase sleep1 started")
PrintToRMSLog ("Warning", "TestCase 1 started a second time")
```

Definition of user function in `rms.inc`:

```
PrintToRMSLog (STRING sMessageType, STRING sUserMessage)
```

Writes to the log file in the following format:

U | {sTestCaseName} | {sScriptName} | {sArgStr} | {sUserMessage} | {sMessageType} 

Issue Tracking and Results Analysis

Analyze your test results, create and track issues that have occurred, and use reports to document the results.

Analyzing Test Runs

This section describes how you can analyze test runs with Silk Central.

Analyzing Manual Test Results

When you have a test run in either a finished or unfinished context, use the **Manual Test Results** view to provide an easy to read result view for manual tests including all information like attached issues and attachments. All of this information is immediately visible. You can print and download a PDF version as well.

The **Manual Test Results** view contains a tree view on the left and a details view on the right. The tree view is based on context of the selection in which the result view was opened. For example, if you selected a test cycle, you could see all the assigned testers and their tests. Selecting a specific node updates the details view with the appropriate information of the selection.

To open the **Manual Test Results** view, click the **View Manual Test Results** icon () in a **Runs** grid, in a dashboard panel, or in one of the activities pages.

Analyzing Automated Test Results

View the test results of an execution plan run for an enhanced root cause analysis or compare the results of multiple runs to visualize distinctions of test statuses.

The **Test Results** view and the **Run Comparison** view provide an easy to read results overview with full drill-down capabilities. All information is included, like result files, thumbnail screenshots, messages, and stacktraces. This allows you to quickly analyze the root cause of failed test runs, especially because you can also compare the results of multiple runs side-by-side.

For example, you may be interested in performing the following tasks:

- Analyze the runs of all configurations within a configuration suite or within execution plans which represent configurations
- Compare runs of the same execution plan, for example to find out if any test statuses have changed
- Show all failed tests of an execution plan at once

To analyze the results of automated test runs:

1. In the execution **Runs** grid or the **Activities > Last Executions** grid, select one or more executions.
2. Click the **View automated test results** icon to analyze the results of a single execution, or right-click your selection and choose **Compare Automated Test Results** to compare multiple executions.
3. Click the icon at the top of the page or a specific status icon to access detailed information like result files, thumbnail screenshots, messages, and stacktraces.



Tip: Use the **Show failed tests only** check box to narrow down the displayed results.

Comparing Test Results of Multiple Test Runs

Compare the results of multiple runs of one test to visualize distinctions of test statuses.

The **Test Results** view provides an optimized results overview with full drill-down capabilities. All information is included, like result files, thumbnail screenshots, messages, and stacktraces. This allows you to quickly analyze the root cause of failed test runs, especially because you can compare the results of multiple runs side-by-side.

1. In the **Test Runs** grid (**Tests > Details View > <Test> > Runs**), select two or more runs.
2. Right-click your selection and choose **Compare Automated Test Results** to compare multiple test runs.
3. Click the icon at the top of the page or a specific status icon to access detailed information like result files, thumbnail screenshots, messages, and stacktraces.



Tip: Use the **Show failed tests only** check box to narrow down the displayed results.

Changing the Status and Build of a Test Run

To manually change the status of a test run:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan that you want to edit.
3. Click the **Runs** tab.
4. Select the execution plan run. The test section of the **Runs** page lists the test runs.
5. Click on the **Run ID** of the test.
The **Test Run Results** dialog box displays.
6. Click the **Details** tab.
7. Click **Change Status and Build**. The **Change Status and Build** dialog box opens.
8. Select the new status for the test run from the **New Status** list box.
9. Select the new version for the test run from the **Version** list box.
10. Select the new build for the test run from the **Build** list box.
11. Type an explanation for the manual status change in the **Comment** field.



Note: Inserting a comment is mandatory.

12. Click **OK** to confirm the status change.



Note: Status and build changes produce history changes. To view the history of all status changes for the test run, click the **Messages** tab in the **Test Run Results** dialog box.

Viewing Test Run Details

To view the details of a test run:

1. In the menu, click **Execution Planning > Details View**.
2. Select an execution plan in the **Execution Plans** tree.
3. Click the **Runs** tab.
4. In the **Test Runs** grid at the bottom, click the **Run ID** of the test for which you want to see details. The **Test Run Results** dialog box appears.
5. Click the **Details** tab.

Deleting Test Run Results

To delete the results of a specific test run:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution Plans** tree, select the execution plan that you want to edit.
3. Click the **Runs** tab.
4. In the **Actions** column of the test run for which you want to delete results, click .
5. Click **Yes** on the subsequent confirmation dialog box to complete the deletion.

Deleting Runs and Result Files of Execution Plans

You can decide if you want to delete runs (including result files as well as all other items and information that belong to the run) or if you want to delete just the result files of the runs. Result files can be files that require a lot of storage in the database, like videos or screen images. By deleting just the result files, you can clean up your database and free up storage space, but at the same time you keep all the essential information about your runs.

To delete runs or result files:

1. In the menu, click **Execution Planning > Details View**.
2. Right-click on an execution plan, a folder, a configuration suite, or a project in the **Execution Plans** tree and select **Delete Runs or Result Files**.
3. The **Delete Runs or Result Files** dialog box appears. Select what you want to delete:
 - Click **Run result cleanup rules** to automatically cleanup results and result files for all execution plans in all subtrees of the selected execution plan, folder, configuration suite, or project based on the specified result cleanup rules. For additional information, see [Automatically Deleting Results and Result Files](#).
 - Click **Delete all finished runs (including result files)** to delete all runs with all result files.
 - Click **Delete result files of all finished runs** to delete just the result files but not the runs themselves.
4. Check **Within the time span from ... to ...** to delete just the runs or result files of a certain time period.
5. Check **Keep last run** to delete all runs (within the time span, if specified) except the last run.
6. Click **OK**.



Note: Runs of tagged builds will not be deleted with this action. To delete runs of tagged builds, untag the build in **Administration > Products, Version, and Builds > Products**.



Note: If there are many items to be deleted, there may be a slight delay for the UI to update while the items are removed from the database.



Note: If you right-click the project node to delete all runs or result files of the project, the runs or result files of all unassigned tests will be deleted as well.

Execution Plan Run Results

Tracking > Activities > Last Executions > Run ID

The **Execution Plan Run Results** dialog box lists the details of an execution plan run.

You can also access the dialog box from **Execution Planning > Details View**. Select the execution plan for which you want to see details, click the **Runs** tab, right click on the run and choose **View details**.

The dialog box shows detailed information about the following items:

| Item | Description |
|---------------------|--|
| Execution Plan Name | Name of the execution plan. |
| Execution Plan ID | Unique identifier of the execution plan. |

| Item | Description |
|---------------------------------|---|
| Execution Plan Run ID | Identifier of the execution plan run. |
| Start Time | Time the run was started. |
| Duration | Time consumed to execute all included tests. Includes the duration of the setup and cleanup test and the time consumed to fetch automation files and code coverage, launch execution tools, and other tasks. For manual executions this is the time between starting and finishing the execution. |
| Execution Server | Execution server assigned to the execution plan. |
| Warnings/Errors | Amount of warnings and errors generated during the run. |
| Status | Displays the number of passed, failed and not executed tests in a bar graph. |
| Version/Build | Version and build of the product specified for the run. These either are set by the user for this execution plan, or are read from the build information file at execution time. If the build is tagged, it is marked bold and with  . |
| Silk Test Classic AUT Host Name | Name of the Silk Test Classic AUT (Agent Under Test) Host. |
| Setup Test | The test that prepared the testing environment in anticipation of the test. Click on the name of the test to view or edit it. Click on the ID of the test run to open the Test Run Results dialog box. |
| Cleanup Test | The test that restored the testing environment to its original state following the test. Click on the name of the test to view or edit it. Click on the ID of the test run to open the Test Run Results dialog box. |

The **Execution Plan Run Results** dialog box provides additional information about the files included and the messages generated during the execution plan run. It also lists all the assigned tests for the execution plan.

For manual tests click **Manual Test Results** to get a read-only version of the **Current Runs** page, with detailed information on the manual test.

The **Assigned Tests** section lists all tests that are assigned to this execution plan. Click on the name of a test to view or edit it, or click on the **Run ID** of a test to open the **Test Run Results** dialog box.

Test Run Results

The **Test Run Results** page lists run details of a test or execution plan. You can access the page from the following locations:

- **Tests > Details View > <Test> > Runs > <Run ID>**
- **Execution Planning > Details View > <Execution Plan> > Runs > <Run ID>**
- **Execution Planning > Details View > <Execution Plan> > Runs > Test Runs > <Run ID>**
- **Tracking > Activities > Last Executions > <Run ID> > Assigned Tests > <Run ID>**

The **Test Run Results** page displays the following tabs/sections:

| Tab | Description |
|---------------------------|--|
| Details | <p>Shows the details of the run, including its Duration, Execution Path, the Execution Plan Run ID of the execution plan run that included the test run, any Warnings/Errors, and the run comment, if a comment is added to the run. This tab also allows you to change the status of the test run. This option is useful if you need to manually overrule the status of a test run.</p> <p>When a manual status change is performed, the details of the change are reflected in this tab's Status, Status Changed On, Status Changed By, Previous Status, and Status Change Comment fields.</p> |
| Timeline | <p>The Timeline tab shows a grid that allows easy result and error analysis of test runs by combining the following:</p> <ul style="list-style-type: none"> • The structure of the test suite if an <code>output.xml</code> file is available including failure/error/warning count. • Incidents from the <code>output.xml</code> file. • Log entries and messages. • Result files. <p>The entries are listed chronologically and the view contains check boxes allowing you to filter the different types of items.</p> <p>You can also access this tab from the Analyze Results and Analyze Errors actions in the Test Run grids.</p> <p> Note: The tab is displayed for all test types except for manual tests and Silk Test Workbench tests.</p> |
| Specific | <p>Only displayed for Silk Test Classic, Silk Performer, and manual tests. This tab includes details that are specific to the selected test type. For example, when a Silk Test Classic test is selected, this view includes the selected test case, test data, and any warnings that were displayed during the test run.</p> |
| Files | <p>Lists all files that were generated by this run, along with file sizes. The names of Silk Test Classic <code>.rex</code> files act as download links. Once downloaded, these files can be viewed directly in a text editor.</p> <p>The upper table lists files that are associated with the test, such as result files or manually uploaded files for manual tests. The lower table lists files that are associated with the execution plan, for example execution log files or code analysis results.</p> <p>In the Actions column of the Result Files, click  to delete a specific result file. Note that some files are protected and cannot be deleted.</p> <p>Click Download All Files to download all result files generated by the test run, as a zipped package.</p> <p>For Silk Test Workbench, click the <code>result.stwx</code> file to open Silk Test Workbench in context of the result file.</p> <p>For Silk Test Workbench, click the <code>error.png</code> to see the last screenshot for a Visual Test when a playback error occurs.</p> |
| Messages | <p>Lists all messages that were generated by this run, along with the timestamp and severity of the messages.</p> |
| Success Conditions | <p>Only displayed for automated tests. This tab shows all the success conditions that are defined for the test in Tests > Details View > <Test> > Properties during the</p> |

| Tab | Description |
|--------------------|--|
| | test planning process, and the result values from the execution run. Success conditions are used to determine if a test is Passed or if it has Failed . |
| Data Driven | Only displayed for data-driven tests using the option of having a single test for all data rows of the data set. This tab lists the status of each instance (data row) run of the test. Clicking an instance brings up another instance of the Test Run Results dialog box with run details of the selected instance. |
| Attributes | Any attributes that are configured for the test. |
| Parameters | Any parameters that are configured for the test. |

The following table lists the UI elements that are used to step through the test results of an execution run. These elements are only visible when accessing the **Test Run Results** dialog box from an execution plan.

| Item | Description |
|-----------------------------|---|
| Skip Passed | Used to determine which test run results should be displayed when browsing using the Previous Result and Next Result buttons. Checking this option only displays tests with a status other than Passed . |
| < Previous Result | Jumps to the result details of the previous test in the selected execution plan run. |
| Next Result > | Jumps to the result details of the next test in the selected execution plan run. |

Viewing Execution Activities for Data-Driven Tests

To view execution activities for data-driven tests:

1. In the menu, click **Execution Planning > Details View**.
2. Select an execution plan that is based on a data-driven test.
3. In the menu, click **Tracking > Activities**.
4. Click the **Run ID** of the relevant execution plan.
5. In the **Assigned Tests** table, click the **Run ID** of a data-driven test.



Note: If you are running a multiple data-driven test, you will see one test for each data row in your data source.

The results page for that particular test opens.

6. Click the **Data Driven** tab. Here you can view all instances of the test that were executed.



Note: The test's data-driven properties are listed on the **Details** page in the **Data-driven Properties** table.

7. Click an instance name to view test run details for that specific instance.



Note: If you are working with multiple data-driven test instances, a separate instance will be created for each data row in your data source.

8. Click the **Parameters** tab to view the data source values that were used during this specific test run.

Execution Runs Page

Execution Planning > Details View > <Execution Plan> > Runs

The **Runs** page shows statistics regarding all the runs of the selected execution plan.

For configuration suites, the page displays all runs of the included configurations.

For testing cycles, the page displays all runs of the included execution plans and folders.

The page is split into two separate sections, one listing the execution plan runs, and the second listing the test runs for the execution plan run selected in the first section.

The grids are customizable: You can display or hide columns, adjust the width, and drag the columns around to change their order. Click on the columns to sort, group or filter the content in the grids.

To open the **Execution Plan Run Results** dialog box, right-click on a run and choose **View Details**. To compare multiple execution plan runs, use **Ctrl+Click** or **Shift+Click** to select the runs. Right-click on your selection and select **Compare Automated Test Results**.

The execution plan runs section lists the runs of the selected execution plan. Use the menu bar on the bottom of this section to define how many rows to display per page and to navigate through the pages. The page shows the following columns for each run:

| Column | Description |
|-----------------------|--|
| Actions | Actions that you can perform on the execution plan. <ul style="list-style-type: none"> Delete Runs Click to delete the results of this run. When you delete the results for selected runs, Silk Central removes the runs from the Runs page. The runs are grayed out until the background process completes the deletion. Alternatively, press Delete to delete the test run results. For tagged builds, you have to verify that you want to remove the results. View Manual Test Results (only for manual tests) Click to analyze the results of a single manual test run on the Manual Test Results page. View automated test results (only for automated tests) Click to analyze the results of a single automated test run on the Test Results page. |
| ID | The identifier of the execution plan. |
| Execution Plan | The name of the execution plan that was executed. This column is only displayed for configuration suites. |
| Status | Status summary of the run. A bar lists the amount of passed, failed, and not executed tests. The run status of each assigned test is shown in the second section of the page. |
| Run ID | Identifier of the execution plan run. Click to access the results of the run. |
| Keywords | Keywords assigned to the execution plan. |
| Executed By | Name of the execution server on which the run was executed. For manual tests the name of the person who executed the run is listed. |
| Product | The application under test. |
| Version | The version is either set by the user for this execution plan, or is read from the build information file at execution time. This information can be set in Administration > Products, Versions, and Builds > Products . |
| Build | The build is either set by the user for this execution plan, or is read from the build information file at execution time. This information can be set in Administration > Products, Versions, and Builds > Products . If the build is tagged, it is marked bold and with  . |
| Start Time | Time the run started. |
| First Activity | <ul style="list-style-type: none"> • For non-distributed automated runs, this column displays the date and time when an execution server is assigned to all test runs. • For distributed automated runs, this column displays the date and time when an execution server is assigned to the first test run. |

| Column | Description |
|--------------------------------|---|
| | <ul style="list-style-type: none"> For manual runs, this column displays the date and time of the first manual testing activity. |
| Duration | Time consumed to execute all included tests. Includes the duration of the setup and cleanup test and the time consumed to fetch automation files and code coverage, launch execution tools, and other tasks. For manual executions this is the time between starting and finishing the execution. |
| Start Type | Shows how the test run was started. Manually, through a Web Service, or from a schedule. |
| Starter Name | Name of the schedule, tester, or Web Service user that started the execution. |
| Start Scope | The scope specified in the Run dialog box. |
| Execution Plan Parent | The name of the execution plan parent. |
| Run Comment | You can use this column to add information to the run. To edit the comment, right-click on the run and select Edit Run Comment . |
| Mobile Device Selection | The device filter that was specified on the Deployment page, if the execution plan is configured to run on a mobile device. |
| Used Device | The device that matched the device filter and was actually used to execute the tests, if the execution plan is configured to run on a mobile device. For manual tests, the device that was last selected by a tester in this execution plan is displayed. |
| | If you have assigned attributes to your execution plans, you can select to display them in the grid. Attributes are freely definable characteristics that can be applied to execution plans. You can use attributes for filtering and reporting purposes. |

The **Test Runs** section lists the test runs for the selected execution plan run. Use the menu bar on the bottom of this section to define how many rows to display per page and to navigate through the pages. The page shows the following columns for each run:

| Column | Description |
|----------------------|---|
| Actions | Actions that you can perform on the test run. |
| | <ul style="list-style-type: none"> View or download results If the test, to which the run belongs, is of a test type that generates result files, click on the icons to view or download the result files. New Issue Click to open the New Issue dialog box and create a new issue for the test. Assign Existing Issues Assign issues from an externally-configured issue-tracking system to the test. View Manual Test Results Click to view the Current Run page in read-only mode. |
| Status | Status summary of the run. For a single test a single status is shown. A bar lists the amount of passed, failed, and not executed tests for a test package or suite node. |
| Status Reason | The reason why a specific test run has the status Passed , Failed or Not Executed . |
| Run ID | Identifier of the test run. Click to open the Test Run Results dialog box. |
| ID | Identifier of the test. This column is hidden by default. |
| Test | Name of the test. Click to access the test in Tests > Details View . The icon corresponds to the test type. |
| Version | The version the run was executed against. |

| Column | Description |
|---------------------|--|
| Start Time | Date and time the run started. |
| Duration | Duration of the test run. |
| Executed By | Name of the execution server on which the run was executed. For manual tests the name of the person who executed the run is listed. |
| Issues Found | Displays the amount of issues that are assigned to the test run. When no issues are assigned to the test run, the column is empty. Click on the link to access the issue in the Issues page of Tests > Details View . |
| Errors | Number of errors that occurred in the run. |
| Warnings | Number of warnings that occurred in the run. |
| Build | The build the run was executed against. |
| Run Comment | For Silk Performer test runs, Silk Performer uses this column to add information to the test run when uploading results. For all other test types, you can use this column to add information to the run. To edit the comment, right-click on the run and select Edit Run Comment . |

Tracking

The **Tracking** area provides project-level information including **Activities**, **Cross-Project Activities**, **Project Overview Report**, and **Quality Goals**.

Activities

This section explains how to manage upcoming, current, and recently-executed test runs.

Deleting Last Executions Runs

To delete a run from the **Last Executions** list:

1. In the menu, click **Tracking > Activities**.
2. In the **Last Executions** area of the **Activities** page, right-click the test run you want to delete and select **Delete Run Results**.
3. Confirm the deletion by clicking **OK**.

Displaying or Hiding Columns on the Activities Page

To display or hide columns on the **Activities** page:

1. In the menu, click **Tracking > Activities**.
2. Open a column header's context menu.
3. Expand the **Columns** submenus to view all the columns that are available in the project.
4. Select the check boxes of all the columns that you want to have displayed. Your column-display preferences will be saved and displayed each time you open the active project.

Entering Issues From the Activities Page

New issues can be entered directly on the **Activities** page.

To enter an issue from the **Activities** page:

1. In the menu, click **Tracking > Activities**.

2. In the **Last Executions** area, click the **Run ID** of the relevant execution plan to view test-execution results.
Each test associated with the execution run is listed in the **Assigned Tests** table at the lower part of the view.
3. In the **Actions** column of the test to which you want to associate the issue, click **New Issue**.
4. Proceed with defining the issue.

Filtering Test Runs on the Activities Page

You can filter the views on the **Activities** page based on column values. You can specify filter strings to be applied to:

- Text-based data fields.
- Calendar filters (using **Before**, **After** or **On** operators) for date-based fields.
- Numerical operators (**>**, **<**, and **=**) for number-based fields.

Filtering Text-Based Values on the Activities Page

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the text-based column that the filter is to be based on.
3. Expand the **Filter** submenu on the context menu to display the **Filter** field.
4. Enter a text string into the field.
5. Press **Enter**. All entries that match the filter criteria are dynamically displayed in the filtered list.

Filtering Date-Based Values on the Activities Page

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the date-based column that the filter is to be based on.
3. Hold your cursor over **Filter** on the context menu to display the **Before**, **After**, or **On** submenu.
4. Choose from the following:
 - a) Hold your cursor over **After** to define a date before which (and including) all entries should be excluded.
 - b) Hold your cursor over **Before** to define a date after which (and including) all entries should be excluded.
 - c) Hold your cursor over **On** to exclude all entries except those that have the specified date.The calendar tool displays.
5. Select a date using the calendar tool (or click **Today** to specify today's date).



Tip: You must explicitly click a date on the calendar tool or press **Enter** to activate date-based filtering changes.

All entries that match the filter criteria are dynamically displayed in the filtered list.

Filtering Number-Based Values on the Activities Page

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the number-based column that the filter is to be based on.
3. Expand the **Filter** submenu on the context menu to display the **> (greater than)**, **< (less than)**, and **= (equals)** operators.
4. Choose from the following:
 - a) Enter a number in the **>** field to define a number less than which (and including) all entries should be excluded.
 - b) Enter a number in the **<** field to define a number greater than which (and including) all entries should be excluded.

c) Enter a number in the = field to exclude all entries except those that have the specified number.



Note: Number values are rounded to two decimal places.

5. Press **Enter**. All entries that match the filter criteria are dynamically displayed in the filtered list.

Filtering Boolean Values on the Activities Page

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the boolean-based column that the filter is to be based on.
3. Expand the **Filter** submenu on the context menu to display the available values.
4. Click one of the **Yes** or **No** option buttons. All entries that match the filter criteria are dynamically displayed in the filtered list.

Filtering Values Using a Predefined List on the Activities page

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the column that has a predefined filter value that the filter is to be based on.
3. Expand the **Filter** submenu on the context menu to display the **Filter** field.
4. Check the check boxes of the filter values that you want to use. All entries that have one of the selected criteria will be displayed.

Grouping Test Runs on the Activities Page

Beyond simply sorting by column, you can chunk entries into groups to facilitate viewing. Groups are based on commonly shared values within the column that grouping is based on.

Applying Groups to Test Runs on the Activities Page

To group entries on the **Activities** page:

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the column that the sort is to be based on.
3. Select **Group by This Field**.

Entries are then organized into groups based on commonly-shared values within the column you have selected.

Removing Grouping from Test Runs on the Activities Page

To remove grouping:

1. In the menu, click **Tracking > Activities**.
2. Open any column header's context menu.
3. Uncheck the **Show in Groups** check box.

Removing Activities Filters



Note: Hiding a column removes all filters that have been applied to the column.

To remove one or more filters:

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the column that has the filter you want to remove.



Note: You can identify filtered columns by their titles, which are displayed in bold, italic text.

3. Do one of the following:
 - a) To remove a specific filter: Uncheck the **Filter** check box.
 - b) To remove all filters: Select **Reset Filters**.

Reordering Columns on the Activities Page

To reorder columns on the **Activities** page:

1. In the menu, click **Tracking > Activities**.
2. Select the column header of the column you want to move.
3. Drag the column to the desired position and release it.

Your column-order preferences will be saved and displayed each time you open the active project.

Resizing Columns on the Activities Page

To adjust the width of columns on the **Activities** page:

1. In the menu, click **Tracking > Activities**.
2. Select the vertical column-header divider of the column you want to adjust.
3. Drag the column boundary to the desired position and release it.

Your column-width preferences will be saved and displayed each time you open the active project.

Restoring Default Activities Page View Settings

Restoring default view settings resets all user-defined settings, which are column order, column width, shown/hidden columns, applied filters, sorting, and grouping, for the current project.

To restore default view settings:

1. In the menu, click **Tracking > Activities**.
2. Open any column header's context menu.
3. Select **Reset View**.

Sorting Test Runs on the Activities Page

To sort test runs on the **Activities** page:

1. In the menu, click **Tracking > Activities**.
2. Open the column header's context menu of the column you want the test runs to be sorted by.
3. Select **Sort Ascending** to have the test runs sorted in ascending order or select **Sort Descending** to have the test runs sorted in descending order. Your sort preferences will be saved and displayed each time you open the active project.

Activities Page

Tracking > Activities

The **Activities** page offers a centralized location from which you can manage upcoming, current, and recently executed test runs for a project. The grid views on the **Activities** page offer filtering, sorting, and grouping options that are configurable for each user separately. You can display or hide columns, adjust the width of columns, and move columns around using click and drag.

The **Activities** page is split into three sections: **Next Executions**, **Current Executions**, and **Last Executions**. The grid views can be resized by dragging the separators between the views.

Context-sensitive menu commands are available for each test run. These commands enable you to link directly to listed execution plans, continue manual tests, manage test-run results, and more.

The **Activities** page makes it easier to identify match points between execution plans and to find specific execution plan information. Standard Windows keyboard shortcuts can be used to select test run entries, making it easy to select and manipulate specific sets of execution plans and test results. Sorting, grouping, and filtering functions are available through context-menu commands to help you better organize and group test runs. All of your view-customization preferences are saved along with your project and will be available to you each time you visit the **Activities** page.



Note: Data on the **Activities** page is not automatically refreshed. Click **Reload** near the paging buttons at the lower part of each view to refresh the page's contents.



Note: You can use **Ctrl+Click** to select multiple queued executions and abort them all with one click.

Next Executions

The **Next Executions** view lists the execution plans that are scheduled to run in the future. To enhance performance when you have numerous execution plans, only the upcoming 50 execution plans are displayed. You can access additional future execution plans by using the available filtering features. To edit an execution plan, right-click the execution plan and choose **Go to Execution Plan** or click on the arrow to the left of the name of the execution plan. This takes you to the **Executions** area where you can view and edit the details of the execution plan.

By default, all execution plans are sorted by **Start Time**. Columns in the **Next Executions** view can not be sorted or grouped.

Next Executions view can be collapsed and expanded by clicking the double-arrow button on the right-hand side of the view's title bar.

For each execution plan, the **Next Executions** view displays the following columns:

| Column | Description |
|------------------------------|---|
| ID | Identifier of the scheduled execution plan. This column is hidden by default. |
| Execution Plan/Folder | Name of the scheduled execution plan or folder. |
| Keywords | Keywords that are assigned to the scheduled execution plan. |
| Manual Testers | For manual tests, this column contains the name of the manual tester that is assigned to the test. This column is blank when no manual testers are assigned to the test. |
| Priority | The priority that has been assigned to the execution plan. |
| Start Time | Scheduled start time of the test run. |
| Execution Plan Parent | The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed. |

Current Executions

The **Current Executions** view lists the execution plans that are currently running (both automated and manual test runs).

To abort an execution plan that is currently in progress, click **Abort** in the **Actions** column of the execution plan. To view or edit an execution plan, right-click the execution plan and select **Go to Execution Plan** or click on the arrow to the left of the name of the execution plan. To view the execution's progress, right-click the automated execution plan and select **View Details** or click the execution plan's **Run ID/Task ID** link. For automated tests that run on a mobile device, the details view also allows you to view a live stream of the activities on the device. Live streams are not available for Sauce Labs devices.

As long as a manual test remains open, the corresponding execution plan remains in the list of **Current Executions** with a status of **Pending**. Click **Continue Manual Test** in the **Actions** column to open the **Manual Testing** window.

To view the results of a manual test, right-click the manual execution plan and select **View Details**, or click the execution plan's **Run ID/Task ID** link, to open the **Results for Execution Plan** page. From there, click the name of the manual test in the **Assigned Tests** section to open the **Results** dialog box. Detailed results of the manual test are displayed there. Back on the **Results for Execution Plan** page, click **Manual Test Results** to go to **Manual Test** view, where information about the status of the assigned manual test is available.

Page views of current executions are broken into views of 20 execution plans each. You can advance through pages using **First**, **Last**, **Next**, and **Previous** located in the lower part of the **Current Executions** view. Or you can type a page number into the **Page** field and press **Enter**.

For each execution plan, the **Current Executions** view displays the following columns:

| Column | Description |
|-----------------------|---|
| Actions | <p>You can perform the following actions on the execution plan:</p> <p>Abort Click to cancel the current execution. Alternatively, press Delete. When you abort executions, these executions are grayed out until the background process completes the deletion.</p> <p> Note: If the execution is part of a testing cycle the abort action is disabled. The  icon and the entry in the context menu are grayed out and such runs are ignored when you abort multiple runs.</p> <p>Continue Manual Test Click to open the Manual Testing window.</p> <p>View Manual Test Results Click to view the Current Run page in read-only mode.</p> |
| ID | Identifier of the execution plan. This column is hidden by default. |
| Execution Plan | Name of the execution plan. |
| Run ID/Task ID | Manual tests receive a Run ID when they are executed. When the manual test is completed, the Run ID carries over to the Last Executions view. Automated tests receive a Task ID when they are executed. The Task ID is not carried over to the Last Executions view. Completed execution plans receive a Run ID in the Last Executions view. |
| Status | Status of the active execution plan or manual test. For automated tests, the status is indicated with a text-based value. For manual tests, the status is indicated with a colored histogram. Automated-test statuses are described textually and can be filtered. Manual tests can be filtered by checking relevant properties on the Filter submenu. |
| Keywords | Keywords that are assigned to the execution plan. |
| Executed By | <ul style="list-style-type: none"> For manual tests, this column displays the users who are assigned to perform the manual test. This column is blank when no manual testers are assigned to the test. For automated executions, this column displays the name of the execution server. |
| Version | The version that either is set by the user for this execution plan, or is read from the build information file at execution time. This column is hidden by default. |
| Build | The build that either is set by the user for this execution plan, or is read from the build information file at execution time. If the build is tagged, it is marked bold and with  . |

| Column | Description |
|------------------------------|--|
| Priority | For automated execution plans, when more than one execution plan is queued but only one execution server is available, the Priority determines which execution plan is executed first. The same is true if execution servers are available but multiple execution plans require an unavailable mobile device. |
| Start Time | Time when the execution plan run started. |
| Time Left | Amount of time remaining until the test is complete. For manual tests that do not have an estimated time, this column has a value of <code>unknown</code> . |
| Start Type | Shows how the test run was started. <ul style="list-style-type: none"> • Manually • Through a Web Service • From a schedule • From a dependency |
| Starter Name | Name of the schedule, tester, or Web Service user that started the execution. |
| Start Scope | The scope defined in the Run dialog box . |
| Execution Plan Parent | The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed. If you have assigned attributes to your execution plans, you can select to display them in the grid. Attributes are freely definable characteristics that can be applied to execution plans. You can use attributes for filtering and reporting purposes. |

Last Executions

The **Last Executions** view lists all past execution plan runs. This list can be filtered and sorted, for example by start time.

To view or edit an execution plan, right-click the execution plan run and choose **Go to Execution Plan**, or click on the arrow to the left of the name of the execution plan. To display the **Results for Execution Plan** page of an execution plan run, right-click the run and select **View Details**, or click the **Run ID** link of the execution plan. This page shows details for the selected execution plan run and includes any files and messages that were generated during the execution. To open the **Results** dialog box of a test, click on the **Run ID** of the test in the **Assigned Tests** portion of the **Results for Execution Plan** page.

To compare execution plan runs, use **Ctrl+Click** to select multiple runs. Right click on your selection and click **Compare Automated Test Results** for automated tests, or **Reports > Run Comparison ...** for manual tests.

Right-click on an execution plan and click **Run Execution Plan** to run the execution plan. Silk Central considers all tests that are currently assigned to the execution plan, not the originally assigned tests of the selected execution plan run.

To delete an execution plan run, right-click a run entry and choose **Delete Results** or click **Delete** in the **Actions** column of the run.

Test-result page views are broken into views of 20 test results each. You can advance through pages using **First**, **Last**, **Next**, and **Previous** at the bottom of the **Last Executions** view. Or you can enter a page number into the **Page** field and press **Enter**.

The **Last Executions** view can be collapsed/expanded by clicking the double-arrow button on the right-hand side of the view's title bar.

For each execution plan, the **Last Executions** view displays the following columns:

| Column | Description |
|--------------------------------|---|
| Actions | <p>You can perform the following actions on the execution plan:</p> <p>Delete Click to delete the execution plan run results. When you delete executions, these executions are grayed out until the background process completes the deletion. Alternatively, press <code>Delete</code> to delete the execution plan run. For tagged builds, you have to verify that you want to remove the results.</p> <p>View Manual Test Results Click to view the Current Run page in read-only mode.</p> |
| ID | The identifier of the executed execution plan. Unassigned tests have an ID value of N/A. This column is hidden by default. |
| Execution Plan | Name of the executed execution plan. Click the button beside the name to view the execution plan in the Execution Planning unit. |
| Run ID | ID assigned to the test run. Click the link to view details of the test run. |
| Status | Displays the number of passed, failed and not executed tests in a bar graph. Filtering, sorting, and grouping are not available for the column. |
| Keywords | Keywords that were assigned to the execution plan at execution time. |
| Executed By | <ul style="list-style-type: none"> For manual tests, this column displays the users who are assigned to perform the manual test. This column is blank when no manual testers are assigned to the test. For automated executions, this column displays the name of the execution server. |
| Start Time | Time when the execution plan run started. |
| First Activity | <ul style="list-style-type: none"> For non-distributed automated runs, this column displays the date and time when an execution server is assigned to all test runs. For distributed automated runs, this column displays the date and time when an execution server is assigned to the first test run. For manual runs, this column displays the date and time of the first manual testing activity. |
| Duration | Time consumed to execute all included tests. Includes the duration of the setup and cleanup test and the time consumed to fetch automation files and code coverage, launch execution tools, and other tasks. For manual executions this is the time between starting and finishing the execution. |
| Product | The product under test. This column is hidden by default. |
| Version | The version that either is set by the user for this execution plan, or is read from the build information file at execution time. This column is hidden by default. |
| Build | The build that either is set by the user for this execution plan, or is read from the build information file at execution time. If the build is tagged, it is marked bold and with  . |
| Start Type | Shows how the test run was started. Manually, through a Web Service, or from a schedule. |
| Starter Name | Name of the schedule, tester, or Web Service user that started the execution. |
| Start Scope | The scope defined in the Run dialog box . |
| Execution Plan Parent | The configuration suite, folder, or testing cycle in the context of which the execution plan is executed. Click to access the suite or folder in the Execution Plans tree. If the execution plan is not included in a configuration suite or folder, nothing is displayed. |
| Mobile Device Selection | The device filter that was specified on the Deployment page, if the execution plan is configured to run on a mobile device. |

| Column | Description |
|--------------------|---|
| Used Device | <p>The device that matched the device filter and was actually used to execute the tests, if the execution plan is configured to run on a mobile device. For manual tests, the device that was last selected by a tester in this execution plan is displayed.</p> <p>If you have assigned attributes to your execution plans, you can select to display them in the grid. Attributes are freely definable characteristics that can be applied to execution plans. You can use attributes for filtering and reporting purposes.</p> |

Cross-Project Activities Page

Tracking > Cross-Project Activities

The **Cross-Project Activities** page enables a user with SuperUser privileges to see all execution related activities across projects. It provides all the options the **Activities** page offers, and additional options across projects. This allows some conclusion about the execution queue. The SuperUser can remove executions from the queue to resolve bottlenecks.

The **Cross-Project Activities** page is visible to the SuperUser only. The page is split into the same three sections as the **Activities** page: **Next Executions**, **Current Executions**, and **Last Executions**. All three sections include additional columns with the **Project Name** and the **Project ID**.

| Item | Description |
|--------------|---|
| Project Name | The name of the project to which the execution plan belongs to. |
| Project ID | The ID of the project to which the execution plan belongs to. |

Project Overview Report

Silk Central > Tracking > Project Overview Report

The *Project Overview Report* contains a high-level overview of the status of the selected project.

Silk Central, Risk-Based Testing, and Quality Goals

Often during software development, a balance must be struck between testing scope, time, and cost. One commonly used method in determining this balance is risk-based testing.

A key practice of risk-based testing is the process of categorizing test assets based off of factors important to the quality and to the end users of the application.

Based off of this categorization process, quality practitioners can determine the appropriate set of tests based off of time available and the desired goals of the quality process. Executing a subset of tests involves some amount of risk. Quantifying this risk in a relevant way ensures that the stakeholders are informed throughout the process, which is key for a successful implementation of risk-based testing.

In Silk Central, factors used for risk-based testing are completely customizable using *Quality Goals*. A quality goal can be created for any **List** type custom attribute or requirement property which are used to categorize testing assets. This allows users to define a subset of tests that need to be executed in order for the project to meet specific quality criteria. Quality goal reporting allows users to communicate to stakeholders the test planning and execution process, relevant scope, time, and quality metrics on testing assets.

Quality Goals

Quality goals (exit criteria) are a definable set of testing metrics that the project must meet in order for the quality to be considered acceptable. Each Silk Central project can contain one or more quality goals. Silk Central uses quality goals as a primary means to implement risk based testing, allowing you to plan, to test, and to report on only the most important tests in a given testing cycle.

In Silk Central, a quality goal is comprised of the following two elements:

- A requirement or test.
- One of the following:
 - A custom attribute or property of type **List** with a corresponding percentage value.
 - One of the following default requirement properties: **Priority**, **Reviewed**, or **Risk** with a corresponding percentage value.

Quality Goal Example

If a requirement has a property of `MyCustomRisk` with values of `High`, `Medium`, and `Low`, it should be possible to define a quality goal of:

Requirement Goal - MyCustomRisk

High = 100%

Medium = 50%

Low = 10%

The 100% means that all of the tests that meet that criteria (linked to a requirement that has the *High* attribute value) must execute and pass to meet that goal.

Permissions

Quality goals are permission based. See *Quality Goals Permissions* for details of permission.

Quality Goals Page

Tracking > Quality Goals

The following controls are available on the **Quality Goals** tab allowing you to create, edit, or delete quality goals for a project:

Quality Goals Grid

| Column | Description |
|------------------|---|
| Goal Name | The name of the goal. Select the drop-down list on the column name and then select Show in Groups to show all goals in a list. |
| Name | Each value of the attribute or property <code>List</code> type is itemized in the Name column. |
| Goal (%) | This text field must contain a numeric value from 0 to 100. The value represents the percentage of tests required to meet the defined quality goal. |

Buttons

- **Manage Goals** - Click to show the **Manage Quality Goals** dialog box.
- **Planning Report** - Click to show the **Quality Goals Planning Report**.

- **Execution Report** - Click to show the **Quality Goals Execution Report**.

Adding Quality Goals

Before creating a quality goal, you need create a test custom attribute or requirement custom property of type `List` that you want to use for your quality goals. Or you can use the requirement fields of **Priority**, **Reviewed**, or **Risk**.

For example, if you want a quality goal to track a requirement custom property with values of `High`, `Medium`, and `Low`, first go to **Projects:<Project Name> > Project Settings > Requirement Properties** and create a `List` type custom property that contains these values.



Note: You can only use a property or attribute as a quality goal once per project.

1. In the menu, click **Projects > Project List**.
2. Select a project.
3. Click **Tracking > Quality Goals**.
4. Click **Manage Goals**. The **Manage Quality Goals** dialog box opens.
5. In the **New Goal Name** field, enter the name for your goal.
6. In the **Attributes/Custom Properties** list, select the item that contains the values that you want to use for your quality goal. The list contains the following types of items:
 - Requirement property of type **List**.
 - Default requirement properties of **Priority**, **Reviewed**, or **Risk**.
 - Test attribute of type **List**.
7. Click **Add New Goal**.
8. Click **Close**. The **Manage Quality Goals** dialog box closes and the new goal is added to the **Quality Goals** grid.
9. For each value in the list, enter a numeric value from 0 to 100 in the **Percentage of Tests Required to Meet Goal**.

Deleting Quality Goals

To delete a quality goal:

1. In the menu, click **Projects > Project List**.
2. Select a project.
3. Click **Tracking > Quality Goals**.
4. Click **Manage Goals**. The **Manage Quality Goals** dialog box opens.
5. Click the **Delete** icon next to the quality goal that you want to delete.

Implement a Risk-Based Testing Approach by using Quality Goals

For a given project, a user wants to balance the scope of testing and time while still ensuring the application is of acceptable quality. The following steps illustrate a high-level workflow of how a user can use quality goals to implement a risk-based testing approach.

In this project, the stakeholders are interested in ensuring enough of the test bed has been executed as well as ensuring the most important requirements have been sufficiently tested. The data for the below example is also included in the `Demo` project that is included with new versions of Silk Central.

1. Select a project.
2. Create a requirement property of type **List** to capture the importance of the requirement.

For this example, create the requirement property `Business Value` with values:

- Low
- Medium
- High

3. Create a test attribute of type **List** to capture the type of purpose of the tests.
For this example, create the test attribute `Level` with values:

- Full
- Regression
- Smoke

4. Click **Tracking > Quality Goals**.

5. Create a new quality goal.

For this example, create the quality goal `Business Value` for requirement property with the following **Goal %**:

- High-100%
- Medium-75%
- Low-50%

6. Create a new quality goal.

For this example, create the quality goal `Test Objectives` for the test attribute with the following **Goal %**:

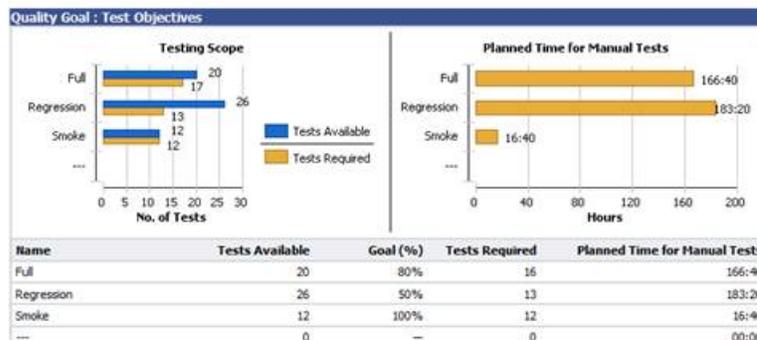
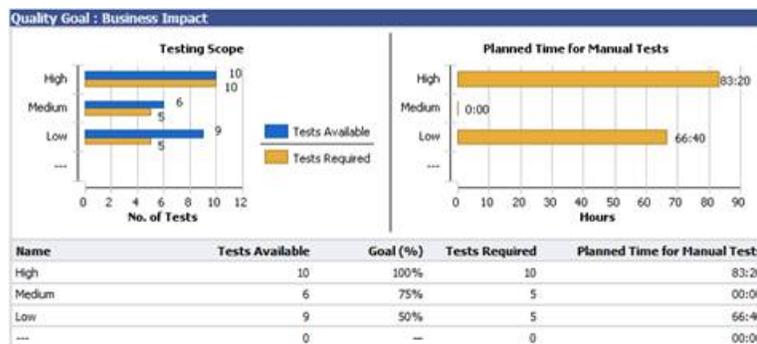
- Full-80%
- Regression-50%
- Smoke-100%

7. After the test assets are developed, assign the attributes/properties to the requirements and tests.

8. To determine the testing scope and ensure stakeholder agreement in the testing scope, click **Reports > Details View**.

9. Click **Tests > Status Report > Quality Goals Planning Report**.

This report will show you the number and percentage of tests for each quality goal value, total planned time for manual tests, and the number of tests required to meet the goal vs. the number of test available in the project.



10. After agreement on the plan has occurred and the project is progressing through testing execution, testing status for the Quality Goals can be easily evaluated using the **Quality Goals Execution Report**.

11. Click **Tests > Status Report > Quality Goals Execution Report**.



Issues

Describes how to work with issues in Silk Central.

Creating New Issues

The **Issues** page enables you to easily create issues related to the selected test.

To create a new issue:

1. In the menu, click **Tests > Details View**.
2. Select the test for which you want to create a new issue.
3. Click the **Issues** tab.
4. Click **New Issue** to open the **New Issue** dialog box.
5. Select the **Profile** of the issue-tracking system you are submitting the issue to.

 **Note:** The profile you select here becomes the default selection for when you enter new issues in the future. When adding a new issue to an issue tracking system, you will be prompted to provide login credentials for the issue tracking system. The credentials that you provide will be automatically preselected for you in the future.

6. Enter a brief **Synopsis** of the issue.
7. Enter a meaningful **Description** of the issue.
8. Provide information about the issue by using the UI controls on the right side of the dialog box.

 **Note:** These UI controls vary, based on the selected **Profile**. For example: If you want to create an issue that shall be processed with Bugzilla, you need to specify the product, the component, and so on. If an issue shall be processed with JIRA, you need to specify other values, like the issue type or the priority.

9. Click **OK** to create the issue.

Viewing Issue Statistics in Document View

To view issue statistics in **Document View**:

1. In the menu, click **Issues > Document View**.
2. In the **Issues** tree, select the project, issue-tracking system, or product for which you want to view statistics.

Issues Document View

Issues > Document View

The **Document View** displays issue statistics for the selected project in tabular format.

| Column | Description |
|-----------------|--|
| Date | Date and time when issue details were updated. |
| Open | Number of issues in the selected project, database, or product that have a status of Open . |
| Fixed | Number of issues in the selected project, database, or product that have a status of Fixed . |
| Verified | Number of issues in the selected project, database, or product that have a status of Verified . |
| Closed | Number of issues in the selected project, database, or product that have a status of Closed . |
| Deferred | Number of issues in the selected project, database, or product that have a status of Deferred . |

Viewing Issue Statistics in Details View

To view issue statistics in **Details View**:

1. In the menu, click **Issues > Details View**.
2. In the **Issues** tree, select the project, issue-tracking system, or product for which you want to view statistics.
3. The calendar tool feature enables you to specify the time period over which you want to view issue statistics. Click the time-frame dates link to expand the calendar.
4. Using the **Edit** icons next to the **From** and **To** fields, specify start and end times for issue statistics.
5. Click **Update** to update the chart view based on the specified time range.

Issues Page

Issues > Details View > Issues

The **Issues** page lists the issues from all issue tracking systems that are configured for the selected project. The page includes the following items:

| Item | Description |
|---------------|---|
| Calendar Tool | Allows you to specify a time-frame for which issues should be reported. Click the time-frame date link to expand the calendar tool. |
| Update | Updates the Issues View based on calendar changes. |

Assigning Existing Issues

The **Issues** page enables you to assign issues from issue tracking systems to a selected test.

To assign an existing issue to a test:

1. In the menu, click **Tests > Details View** .
2. Select the node of the test for which you want to assign an issue.
3. Click the **Issues** tab.
4. Click **Assign Existing Issues** to open the **Assign Existing Issues** dialog box.
5. Select the profile of the pre-configured issue-tracking system where the issue is tracked.
6. Type the issue numbers in the **Issue ID** field, separated by commas.
7. Click **OK**.



Note: You can also assign an existing issue to a currently running execution plan in the **Current Run** page, the **Execution Runs** page (in **Execution Planning**), and the **Test Runs** page (in **Tests**).

Updating Issue States

To synchronize issue states between Silk Central and an issue tracking system:

1. In the menu, click **Tests > Details View** .
2. Select the node of the test for which you are updating a corresponding issue.
3. Click the **Issues** tab.
4. Click **Update Issue States** to synchronize the state of the issues listed in Silk Central with the corresponding issues in the issue tracking system.

Deleting Issues (Issue References)

To delete an issue reference:

1. In the menu, click **Tests > Details View** .
2. Select the test from which you want to delete an issue.
3. Click the **Issues** tab.
4. In the **Actions** column, click .
5. Click **Yes** on the **Delete Issue** dialog box to confirm the deletion.



Important: This action removes just the issue reference in Silk Central. It does not delete the issue itself in the issue tracking system.

Issue Manager

Welcome to Issue Manager!

Issue Manager is a comprehensive tool for automatically tracking issues. With Issue Manager you can enter, process, and manage issues in a way that models the workflow of your organization.

Issue Manager assists you in managing software bugs, documentation issues, requests for enhancements, and more. It is designed to meet the needs of a wide range of users, including application developers, QA engineers, technical support, documentation professionals, managers, and IT personnel.

Issue Manager uses a technique called *action-driven workflow*. This workflow technology moves an issue through its life cycle automatically, based on user actions and issue states that can be customized to the workflow of your business. From the time an issue is reported until it is closed, Issue Manager handles issue routing automatically whenever a user takes an action on that issue.

Your administrator sets up routing rules and workflow to automatically advance issues between inboxes that are assigned to individual users or groups. Users with special security privileges can override predefined issue routing.

Issue Manager allows you to perform the following tasks:

- Report issues through a Web interface.
- Add, review, reassign, fix, verify, and close issues.
- View detailed information about any issue.
- View user accounts and inboxes.
- Receive email whenever certain changes occur to an issue.
- Retrieve issue information from the database by running queries.
- Produce reports and charts.
- Archive issues.

Getting Started with Issue Manager

As an end-user

Become familiar with the end user's tasks.

Create a sample database before you create and configure your organization's production database. For additional information, see *Databases*.

As an administrator

Before you can work with Issue Manager, you need to set up a database. For additional information, see *Databases*.

Exploring the Sample Database

As an administrator, explore the sample database to get familiar with the features of Issue Manager. To explore the sample database:

1. Launch Issue Manager and log into the sample database as `dhart`, a **Development** user with unrestricted permissions.
2. Enter an issue against an existing product.
3. Save the issue. Issue Manager fills in the **Issue Details** page's automatic fields and the **History** tab.
4. Take an action on the issue and complete the corresponding action dialog box. The **Issue Details** page changes. Take other actions until you feel comfortable with the process.
5. View several user accounts.
 - a) Note which groups the users are assigned to.
 - b) Look at the user's security privileges.
 - c) Note the default Inboxes of the users.
6. Add a group.
 - a) Assign security privileges.
7. Add an Inbox for a new user.

- a) Associate the Inbox with the new group.
 8. Add an account for the new user.
 - a) Assign the new Inbox to the user.
 - b) Assign any remaining security privileges to this user account.
 9. Add a new product, including its routing rules.
 - a) Add one or two releases for the product.
 - b) Associate each release with one or two platforms.
 10. Log out of the system.
 11. Log in as the new user.
 12. To see how an issue is processed through the workflow, enter and save an issue against your new product.
 - a) In the **Assigned To** field the issue has been routed to one of the Inboxes you have specified in your routing rules. The bug is in the **Dev-Ready** state. Select the **History** tab and note the entries there.
 - b) Select the **History** tab and note the entries there.
 - c) Reassign the issue to another developer.
 - d) See how this action is reflected by the **History** tab.
 - e) Set the issue to **Fixed** to advance it to its next state. The **State** field changes to **QA-Ready** and the **Assigned To** field changes to the Inbox specified in your routing rules. Presumably this is the QA engineer inbox.
 - f) Set the issue to **Verify**, which means that a QA engineer has verified that the bug has been fixed. The issue should move to the **Closed** state. Look at the **History** tab again. How has it changed?
 13. Add two components for the new product.
 14. Set up routing rules for the components, specifying different Inboxes than for the entire product.
 15. Enter another issue against a specific product component. Repeat the sub-steps described in step 12 to see how the routing changes.
-  **Note:** The states the issue passes through remain the same because the workflow is the same for all issues.
16. Change a field label on the **Issue Details** page.
 17. Change **Product Code** to **Product**.
 18. Open the **Issue Details** page for an existing issue to view the change.
 19. Add a new field to **Custom Tab 1**.
 20. Open the **Issue Details** page for an existing issue and select the **Custom** tab to see the new field.
 21. Generate a few of the predefined reports and charts that are available in the **Reports** unit.

What You Can Do With Issue Manager

As a complete solution for your issue tracking needs, Issue Manager allows you to:

- *Create new issues:*
- *Import issues:* The Silk Test Assistant enables you to forward the details of Silk Test test cases directly to Issue Manager as the basis for new issues. Likewise, when you discover Silk Central execution results that require attention, you can easily use the results as the basis for new issues.
- *Process issues:* Issue Manager lets you review, reassign, fix, verify, and close issues (privilege vary based on user type).
- *Associate files with Issue Manager issues:* If you or a customer has files that are required for reproducing an issue, you can upload those files to Issue Manager and attach them to the issue. In this way, you can easily locate the files you need to confirm, fix, or verify a bug.
- *View a complete history of every issue in the system:* Issue Manager records every action that users take on an issue. The default issue history entry includes the action taken on the issue, the user who took the action, and notes that explain the action.

- *Interact with workgroups:* Your Issue Manager administrator can set up workgroups within your organization for sharing privileges, permissions, inboxes, and assignments. In this way, you can interact with users on an individual basis and also in groups that share similar job responsibilities.
- *Query the system for issue information that meets your criteria:* You can use predefined queries, construct queries by example, or create advanced queries with SQL.
- *Generate reports and graphs:* Create predefined reports based on issue criteria you specify or create advanced reports with SQL.
- *Define notifications:* Define to be notified when issue statuses or assignments change.
- *Remotely interact with the issue database:* Enter and query issues remotely via the web.
- *Assign tests to issues:* With Silk Central

Using Issue Manager

The Issue Manager Help provides information and guidelines that will help you track issues with Issue Manager to meet the needs of your software development environment.

Overview

This section gives an overview of the Issue Manager UI. This contains the **Issue Tracking** page, the **Issue Details** page, the **Issue Details** tabs, and the **Issue Tracking** toolbar.

Issue Tracking Page

In the menu, click **Issues > Issue Tracking**.

Overview

The **Issue Tracking** page shows the list of issues of your inbox. To view another inbox, select one from the **Show** list in the toolbar. Inboxes can be organized in groups. To show just inboxes of a certain **Group**, select one from the list in the toolbar. You can display or hide columns, adjust the width, and drag the columns around to change their order. Click on the columns to sort, group or filter the content in the grid.

Click  (**View Issue**) in the grid to show the details of an issue on the **Issue Details** page. You can also use the **Up** and **Down** keys on your keyboard to navigate through the list, and press **Enter** to show the details of an issue.

Use the **Show** list to define which issues you want to see:

- **All open issues:** Displays all issues of all inboxes that are not in a terminal state.
- **All issues:** Displays all issues of all inboxes, including closed issues.
- **<inbox name>:** Displays all issues that are currently assigned to the selected inbox.

Toolbar

Click the buttons in the toolbar to perform the following actions:

- Click  (**My Inbox**) to show the list of issues of your inbox.
- Click **New Issue** to add a new issue.
- Click  (**Download as XLSX**) to export the list of issues to an Excel sheet.
- Enter an issue number in the **Go to Issue #** toolbar field and press **Enter** to quickly move to a certain issue.
- Click  (**Open Silk Test Classic Intelligent Assistant**). This icon is only displayed if the `ShowSilkTestAssistant` tag in the `SRFrontendBootConf.xml` file is uncommented and set to `true`. For more information, see *Silk Test Intelligent Assistant*.

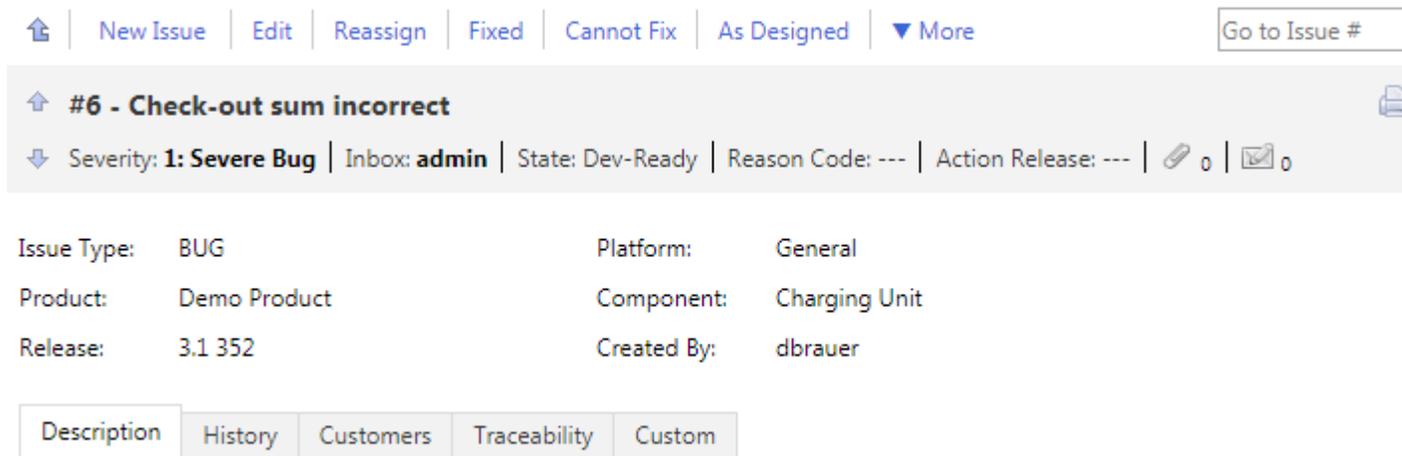
Bulk Updates

You can use your keyboard's **Ctrl** and **Shift** keys to select multiple issues using standard browser multi-select functions. Based on a user's permissions, the following actions can be performed on multiple issues by right-clicking your selection:

- Reassigning issues, if no closed issues are selected.
- Workflow actions, if the selected issues have the same type and state. Note that workflow actions using a product-dependent field are only enabled, if the selected issues have the same product.

Issue Details Page

In the menu, click **Issues > Issue Tracking**, select an **Inbox** from the list in the toolbar, and then click  (**View Issue**) in the grid. The **Issue Details** page shows detailed information about an issue.



The screenshot shows the top toolbar of the Issue Details page. It includes a home icon, a 'Go to Issue #' input field, and a menu with options: New Issue, Edit, Reassign, Fixed, Cannot Fix, As Designed, and More. Below the toolbar, the issue title is '#6 - Check-out sum incorrect'. The metadata bar shows: Severity: 1: Severe Bug, Inbox: admin, State: Dev-Ready, Reason Code: ---, Action Release: ---, and icons for attachments (0) and notifications (0). The metadata table below shows:

| | | | |
|-------------|--------------|-------------|---------------|
| Issue Type: | BUG | Platform: | General |
| Product: | Demo Product | Component: | Charging Unit |
| Release: | 3.1 352 | Created By: | dbrauer |

Below the metadata table is a tabbed interface with 'Description' selected, and other tabs for 'History', 'Customers', 'Traceability', and 'Custom'.

Toolbar

A list of actions is displayed on the toolbar at the top of the page. Click the links to work on a certain issue. The available links vary depending on the state of the issue and on your permissions and privileges. For a list of all available links, see [Actions for Working with Issues](#).

Click  (**Back to Inbox**) to get back to the list of issues on the **Issue Tracking** page. You can also use the **Backspace** key for this.

Enter an issue number in the **Go to Issue #** toolbar field and press **Enter** to quickly move to a certain issue.

Issue Details

Below the action links, the most important information about the issue is displayed, and you can add attachments and configure notifications. Use the **Attachments** and **Notifications** icons below the issue synopsis to add attachments and configure notifications.

| Name | Description |
|--------------------------------|---|
| Previous Issue in Inbox | Shows the data of the previous issue in the list of issues in the selected inbox. You can also use the Up and Left keys for this. |
| Next Issue in Inbox | Shows the data of the next issue in the list of issues in the selected inbox. You can also use the Down and Right keys for this. |

| Name | Description |
|-------------------------------|---|
| Attachments | Allows you to add files with additional information concerning the issue. You can edit and remove the files with the buttons in the Actions column. Does not require to click Edit first. |
| Notifications | Allows you to configure email notifications. Issue Manager notifies you when certain values of an issue change. Does not require to click Edit first. |
| Show printable version | Shows a printable view of the data of the selected issue. |

Issue Details Tabs

The issue details tabs display beneath the issue details on the **Issue Details** page. Before you can edit the values on the tabs (like adding customers or updating the description), you need to click **Edit** on the **Issue Details** page. You can also use the **F2** key for this. The following tabs are available:

| Name | Description |
|---------------------|--|
| Description | Contains a Synopsis and a Description of the issue.  Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for Description fields. |
| History | Tracks all changes concerning the issue. For example: When the issue was created, who created it, actions like reassigning, fixing, verifying, addition of customers, files, testcases, and so on. |
| Customers | Allows you to add information about customers who reported the issue. You can edit and remove the customers with the buttons in the Actions column. |
| Traceability | Shows the tests that are associated with the Issue Manager issue. Click the name of a test to get to the Tests Details View . On the Issues tab of the Tests Details View you can see all issues that were assigned to the test. |
| Custom | Contains any custom fields and lists the Issue Manager Administrator set up. |
| Testcases | Allows you to add Silk Test test cases. QA and development personnel can directly execute the test cases by clicking  (Run Testcase) in the Actions column. Silk Test test cases can also be imported with the Silk Test Intelligent Assistant. This tab is only displayed if the ShowTestcasesTab tag in the SRFrontendBootConf.xml file is uncommented and set to true. |

Entering Issues

This chapter describes how to create and edit issues in Issue Manager, how to use the Silk Test Intelligent Assistant to create issues out of Silk Test result files and how to export results from Silk Test.

Creating Issues (Issue Manager)

To create a new issue in Issue Manager:

1. In the menu, click **Issues > Issue Tracking**.
2. Click  (**New Issue**) in the toolbar.
3. Select values for the **Issue Type**, the **Product**, the **Release**, the **Platform**, the **Component**, and the **Severity**.

 **Note:** The highlighted fields and lists are required.

As you enter values, Issue Manager dynamically assigns an inbox and an initial state for the issue based on configured rules.

4. Enter a **Synopsis** and a **Description**. To create a link between this issue and another issue, type the issue number or the synopsis of the related issue into the **Linked Issue** field (autosuggest-enabled). Linked issues are listed in the **History** tab.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. *Optional:* Add further details to the issue. Click the tabs and click the following buttons:
 - **Add Customer:** Track information about customers who have reported the issue.
 - **Add File:** Add a file with additional information about the issue.
 - **Add Testcase:** QA and development personnel can directly execute the added Silk Test testcases by clicking  (**Run Testcase**) in the **Actions** column on the **Testcases** tab. Silk Test testcases can also be imported with the Silk Test Intelligent Assistant.
 - **Add Notification:** Configure to be notified by email when certain values of an issue change.
6. Click **OK**.

The **Issue Details** page displays. To get back to the list of issues of your inbox (**Issue Tracking** page), click  (**My Inbox**).

Editing Issues (Issue Manager)

To edit an issue in Issue Manager:

1. In the menu, click **Issues > Issue Tracking**.
2. Click  (**View Issue**) in the grid. The **Issue Details** page displays.
3. Click **Edit**.
4. Edit the values of the lists and fields.
5. Click the tabs to edit further details of the issue.
6. Click **OK**.

The changes are tracked in the history. Click the **History** tab to view the changes.

Silk Test Intelligent Assistant

The Silk Test Intelligent Assistant analyzes test cases in Silk Test result files (.rex). The assistant then creates test case issues which can be tracked with Issue Manager.



Note: The **Open Silk Test Classic Intelligent Assistant** icon is only displayed if the `ShowSilkTestAssistant` tag in the `SRFrontendBootConf.xml` file is uncommented and set to `true`.

Based on the test case results you can either create new issues or update existing issues. For example: If you run a test case and it fails, the Intelligent Assistant can attach the test case to a new issue. QA engineers can subsequently execute the attached test case directly from Issue Manager and determine whether the issue has been fixed.

You can assign one or more test cases to an issue. However, you cannot assign a test case to more than one issue. The Intelligent Assistant considers two test cases to be the same if they have the same script name, test case name, and arguments.



Note: You can instruct the Intelligent Assistant to consider the same test case run on different platforms to be separate test cases. The Intelligent Assistant will not propose a test case to be fixed until that test case passes on all platforms.

Importing Silk Test Results to Issue Manager

To import the result of Silk Test tests:

1. In the menu, click **Issues > Issue Tracking**.
2. Select an **Inbox** from the list in the toolbar.
3. Click  **(Open Silk Test Classic Intelligent Assistant)** in the toolbar.
4. Browse for a .rex file on your computer.
5. Select a platform from the list to use the platform that the test ran against as a factor in the consolidation logic.
6. Click the check box to include tests that have no errors and are not already associated with an issue and click **Next**.
7. Click the check boxes to select testcases and click one of the following buttons:
 - Click **New Issue** to create a new issue and assign the testcase to it. For more information see *Creating Issues (Issue Manager)*.
 - Click **Existing Issue** and enter an issue number to assign the testcase to it.

Assigned testcases are no longer shown in the list.
8. Click **Next**.
9. Confirm the proposed actions and click **Finish**.

The testcases are assigned to the issues. To run or edit the testcases, click the buttons in the **Actions** column on the **Testcases** tab.

Exporting Results from Silk Test

To export results from Silk Test:

1. In Silk Test, click **Results > Send to Issue Manager**. The **Send Results to Issue Manager** dialog box displays.
2. Select a results file (.res).
3. Click **OK**. The Intelligent Assistant launches. The **Intelligent Assistant** tab displays with the path to the results file already loaded in the **Exported test results (.rex)** field.
4. Click **Next**. The **Associate Tests with Issues** dialog displays.

Follow the steps in *Importing Silk Test Results to Issue Manager*.

Proposed Changes

When you associate testcase results and Silk Test testcases with issues, the Silk Test Intelligent Assistant compares the results and the current information about issues to build a list of proposed changes for issues.

The list includes proposed actions for issues plus the new issue states and reason codes that will result if you confirm the actions. Each proposed action is based on the current state and reason code of each issue, and on the results of the associated testcases.

To create proposed change list, the Silk Test Intelligent Assistant uses the following equation:

[current state and reason code] + [test results] = [special action, new state and reason code]

For example: If the current state of an issue is *QA-Ready* and its reason code is *Fixed*, if a testcase associated with this issue failed, the Intelligent Assistant will propose the *AutoRejectFix* action, which will change the issue's state to *Dev-Ready* and its reason code to *Rejected*. The following equation summarizes this example:

[QA-Ready and Fixed] + [failed test] = [AutoRejectFix, Dev-Ready and Rejected]

Working with Issues

Issue Manager provides a range of actions that allow you to work on issues. You can, for example, mark issues as *fixed* or *implemented*, you can indicate that you *need more info* or that an issue is *no longer an issue*, and so on.

The buttons for these actions display on the right side of the **Issue Details** page. For a list of all available actions, see *Actions for Working with Issues*. The list of issues varies depending on the state of the issue and on the permissions of a certain user. Many actions can be configured and customized by the Issue Manager Administrator.

Actions for Working with Issues

The **Issue Details** page shows a list of actions at the top of the page. Click the links to work on a certain issue. The actions vary depending on the state of the issue and on your permissions and privileges.

You can perform the following actions:

| Action | Description |
|-------------------------|---|
| Accept | Indicates that the enhancement or documentation issue described in the issue has been approved for future development. |
| Add Comment | Allows you to add a comment about the action you are taking on the issue. |
| Add Workaround | Indicates that the issue will not be fixed right away, but that a workaround will be generated. |
| Already Done | Indicates that the issue is already fixed. |
| Arbitrate | Indicates that there is disagreement about the state of the issue. Development, Documentation, or QA need to discuss the issue and its state. |
| As Designed | Indicates that the issue actually is not an issue because the feature works as it was designed to work. |
| Cannot Do | Indicates that a requested enhancement cannot be implemented. |
| Cannot Fix | Indicates that the issue cannot be fixed. That may be the case because it is too risky or because it would be too time-consuming. |
| Confirm as a Bug | Indicates that the issue actually is an issue that needs to be fixed. |
| Copy to Project | Allows you to copy issues to another project. All customers, attached files, and testcases are copied with the issue. Notifications are not copied. Click the button, select a Project from the list, and click OK . The values from the source issue are filled in the fields and lists. Edit the values as required and click OK . |
| Defer It | Allows you to defer an issue to a later release. |
| Edit | Allows you to edit the values in the fields and lists on the Issue Details page and on the Issue Details tab. |
| Fixed | Indicates that the issue is fixed. Specify the build or release in which the issue was fixed. |
| Implemented | Indicates that the enhancement is implemented. |

| Action | Description |
|---------------------------|---|
| Mark as Duplicate | Marks an issue as a duplicate of another issue. Specify the issue number of the matching issue. That issue will remain open. |
| Need More Info | Indicates that Development needs more information about the issue from QA. That may be the case when a bug cannot be reproduced or when the description is not clear. |
| No Longer an Issue | Indicates that the issue is outdated. That may be the case when new functionality or another bug fix resolved the issue. |
| Not a Bug | Indicates that the issue is not a bug, but rather that the reporter of the issue misunderstood functionality. |
| Not Reproducible | Indicates that the issue cannot be reproduced. |
| Reassign | Allows you to assign an issue to a different inbox. |
| Reject | Indicates either: <ul style="list-style-type: none"> • The issue is fixed but cannot be verified and therefore is rejected. • The requested enhancement was not approved. |
| Reopen Bug | Indicates that QA has reopened the issue. That may be the case when an issue was fixed in a previous build and was reintroduced in a later build. |
| Verify | Indicates that the issue is fixed and accepted as fixed. |

Taking Action on Issues

To take action on an existing issue:

1. In the menu, click **Issues > Issue Tracking**.
2. Click  (**My Inbox**) or select a **Group** and an **Inbox** from the lists in the toolbar.
3. Click  (**View Issue**) in the grid. The **Issue Details** page displays. A list of actions is displayed at the top of the page.
4. Click an action link and enter the required values in the dialog box.

For a list of all available actions, see *Actions for Working with Issues*.

Setting Issue Verification Preferences

With the issue verification preferences you configure how issues are verified each time you create a new issue or work on an existing issue.

To set issue verification preferences:

1. In the menu, click **Issues > Configuration**.
2. Click **Preferences**.
3. Select one of the following options:
 - Click **Always use normal routing**: This is the default system routing.
 - Click **Always verify your own issue**: All entered issues are routed back to you for verification.

- Click **Prompt for each new issue**: Determine for each individual case if the issues are routed back to you or if the issues use standard routing.



Note: You need the **Issue Verification Preferences** security privilege to set the preferences. The Issue Manager Administrator can assign this privilege to you. If you do not have this privilege, the option buttons are disabled.

Email Notification

You can configure Issue Manager to notify you whenever certain predefined events occur. These events are:

- **Reassignment**: An issue is reassigned with no change in state.
- **StateChange**: An issue changes its state.
- **AnyChange**: A change is made to an issue that generates a new history entry.

The Issue Manager Administrator can define additional rules for specific events. Once rules are defined, users can setup email notifications. You can define project-wide notifications or notifications for specific issues. For these actions, you need the following security privileges:

| Action | Required security privilege |
|-------------------------------------|------------------------------|
| Defining notification rules | Manage email notifications |
| Defining email notifications | Email notification per issue |
| Defining project-wide notifications | Manage email notifications |

The system administrator has to configure the email server settings. Otherwise email notifications will not work.

Any user can define a basic email notification, regardless of assigned permissions. If you set this basic notification, Issue Manager notifies you when another user makes changes to an issue in your inbox or assigns a new issue to your inbox.

Defining Notification Rules

Issue Manager provides three default notification rules: **AnyChange**, **Reassignment**, and **StateChange**. To define additional notification rules:

1. In the menu, click **Issues > Configuration**.
2. Click **Notification Rules** on the left side.
3. Click **Add Rule**. The **New Notification Rule** dialog box appears.
4. Enter a **Rule Name**, a **Description**, and a **WHERE Clause**. See [Defining Email Notification Rules](#) for detailed information.
5. Check the **Is Active** checkbox to activate the rule.
6. Click **OK**.

The notification rule is added to the list.



Note: You need the security privilege **Manage email notifications** to define notification rules.

Defining Email Notifications

To define email notifications:

1. In the menu, click **Issues > Issue Tracking**.
2. Click  (**My Inbox**) or select a **Group** and an **Inbox** from the lists in the toolbar.

3. Click  (**View Issue**) in the grid. The **Issue Details** page displays.
4. Click the **Notifications** icon.
5. Click **Add Notification**. The **Add Notification** dialog box displays.
6. Enter the required values and click **OK**. The notification is added to the list.

To edit the notification click  (**Edit Notification**), to delete the notification click  (**Remove Notification**) in the **Actions** column. For information on configuring project-wide notifications, see [Managing Project-Wide Notifications](#).

Configuring Basic Email Notification

To configure basic email notification:

1. In the menu, click **Issues > Configuration**.
2. Click **Preferences** on the left.
3. Check the check box **Notify me when users assign issues to me or make changes to issues in my inbox**.



Note: Any user can define a basic email notification, regardless of assigned permissions. The subject line and content of the basic email notification can be configured in the file *SRFrontendBootConf.xml*.

Reports (Issue Manager)

With the highly customizable Issue Manager reports you can transform data into meaningful charts. Issue Manager provides a number of pre-configured reports, which are divided into three categories. These categories display in the **Reports** tree:

- Popular Reports
- Text Search
- Management Reports

Text searches are queries that return report data in tabular format. Report templates are created with BIRT Report Designer, an Eclipse-based open source report tool, or with Microsoft Excel.

If you do not have any data in the selected project or if you are not connected to the correct Issue Manager database, blank reports are generated.

You can create reports using the GUI-based tools or by manually writing SQL code. With this advanced approach there is virtually no limit to how data can be queried and presented in reports.

For more information on how to work with reports, see *Reports*.

Creating Reports

To create an Issue Manager report:

1. In the menu, click **Issues > Reports**.
2. Select a folder in the **Reports** tree.
3. Click  (**New Child Report**) in the toolbar. The **Create New Report** dialog box appears.
4. Enter a **Name** and **Description**.
5. If you want to **Share this report with other users** check this check box.
6. Enter a **Timeout [s]** to define how long Issue Manager waits for report executions to complete.
7. Select a **Default tab** and a **Result category** from the lists.

If you access a report by using the context menu (for example when you right-click on a test), you are directed to the defined **Default tab**.

8. Select a **Selection criteria**, a **Property**, an **Operator**, and a **Value** from the lists.
Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).
9. Click **More** to create an additional query string and select the **AND** or **OR** operator to connect the query strings. Click  (**Remove**) to remove query strings.
10. Click **Next** to configure the columns of the report. You can **Add Columns**, **Remove All**, set the order by clicking  (**Move Up**) or  (**Move Down**), remove a single column by clicking  (**Remove**), set sorting and grouping options, and enter an **Alias**.
11. Click **Finish**.

You can also create reports by manually writing SQL code. Click **Advanced** on the **Create New Report** dialog box and enter or paste your SQL code. For more information, see *Writing Advanced Queries with SQL*. For a list of all available function placeholders, see *SQL Functions for Custom Reports*.

Adding Subreports

To aggregate the results from multiple reports into the currently selected report, you can add subreports. The result columns and rows of the subreport are concatenated to the results of the selected report.

To add a subreport:

1. In the menu, click **Issues > Reports**.
2. Select a report in the **Reports** tree.
3. Click the **Properties** tab.
4. Click **Add Subreport**. The **Add Subreport** dialog box appears.
5. From the **Reports** tree, select the sub-report you want to append to the current report.
6. Click **OK**. Subreports display on the associated report's **Properties** page in the **Subreports** section.

To delete a subreport, click  (**Remove**) in the **Actions** column.

Editing Report Properties

To edit the properties of a report:

1. In the menu, click **Issues > Reports**.
2. Select a report from the **Reports** tree.
3. Click  (**Edit**) in the toolbar. The **Edit Report** dialog box appears.
4. Edit the values as required.
5. Click **Finish**.

For more information on the UI controls of this dialog box, see *Creating Reports*.

Editing Report Parameters

To edit the parameters of a report:

1. In the menu, click **Issues > Reports**.
2. Select a report in the **Reports** tree.
3. Click the **Parameters** tab.
If any parameters are defined for the report, they are listed here.
4. Click **Edit Parameters**. The **Edit Parameters** dialog box appears.
5. Edit the **Label** or **Value** of the listed parameters as required.
6. Click **OK**.

For more information on report parameters, see *Report Parameters Page*.

Issue Manager SOAP API

Issue Manager offers an open API (Application Programmable Interface) to enable access to the Issue Manager repository from a third-party application, for example to submit a defect.

The interface the WSDL is based on is shown below. This interface also contains the documentation for the methods:

```
/**
 * A Project object has a ProjectID and a ProjectName. The ProjectID
 * is used for all further methods
 * for specifying the Issue Manager project you want to work on.
 * @param sessionID
 * @return Array of initialized Issue Manager projects available
 * for the user used for generating the sessionID
 * @throws SQLException
 */
public Project[] getProjects(long sessionID) throws SQLException;

/**
 * @param sessionID
 * @param projectID
 * @return the names of the 3 issue types
 */
public String[] getIssueTypes(long sessionID, int projectID)
throws RemoteException, SQLException, LoginException;

/**
 * @param sessionID
 * @param projectID
 * @return Array of ProductNames that are configured for the
 * provided Issue Manager project
 */
public String[] getProducts(long sessionID, int projectID)
throws RemoteException, SQLException, LoginException;

/**
 * @param sessionID
 * @param projectID
 * @param product
 * @return Array of ReleaseNames that are configured for the
 * provided product
 */
public String[] getReleases(long sessionID, int projectID,
String product) throws RemoteException, MissingValueException,
SQLException, LoginException;

/**
 * @param sessionID
 * @param projectID
 * @return Array of available PlatformNames
 */
public String[] getPlatforms(long sessionID, int projectID) throws
RemoteException, SQLException, LoginException;

/**
 * @param sessionID
 * @param projectID

```

```

* @param product
* @return Array of ComponentNames that are configured for the
provided product
*/

public String[] getComponents(long sessionId, int projectID,
String product) throws RemoteException, MissingValueException,
SQLException, LoginException;

/**
* The list of Severities can be grouped by IssueType or Product.
If the list is grouped by IssueType the
* parameter issueType has to be provided and the product parameter
can be empty. If the list is grouped by
* product issueType can be empty and the product has to be selected.
* @param sessionId
* @param projectID
* @param product
* @param issueType
* @return Array of values that can be used for entering an issue in
the given project when the provided
* IssueType/Product is selected.
*/

public String[] getSeverities(long sessionId, int projectID,String
product,String issueType) throws RemoteException,
MissingValueException, SQLException, LoginException,
NotExistingValueException;

/**
* @param sessionId
* @param projectID has to be a ProjectID delivered by
the method getProjects
* @param issueType has to be a values delivered by
the method getIssueTypes
* @param product has to be a values delivered by
the method getProducts
* @param release has to be a values delivered by
the method getReleases for the selected product
* @param platform has to be a values delivered by
the method getPlatforms
* @param component has to be a values delivered by
the method getComponents for the selected product
* @param severity has to be a values delivered by
the method getSeverities for the selected projectID
and IssueType/Product
* @param synopsis the Synopsis for the new issue
* @param description the Description for the new issue
* @return the Issue Number of the new issue
*/

public int saveNewIssue(long sessionId, int projectID,
String issueType, String product, String release, String
platform, String component, String severity, String
synopsis, String description) throws RemoteException,
MissingValueException, NotExistingValueException,
SQLException, LoginException;

```

Projects (Issue Manager)

Issue Manager uses a project-based workflow. Only those issues that are associated with the active project can be viewed or processed. To access individual issues, you need to select a project. In the menu, click

Issues > Project List and click the name of a project to activate it. The selected project remains active until you select another one.

Initializing Repository Configuration

You need to initialize a project for Issue Manager before you can use it.

Initialized projects display **initialized** in the **Actions** column.

To initialize a project:

1. In the menu, click **Issues > Project List**.
2. Click  (**Initialize repository configuration for project <project name>**) in the **Actions** column. The **Init Repository** dialog box appears.
3. *Optional:* Click **Create sample data** to have sample data created for the database you are initializing.
4. Click **OK** to start the initialization. The **Start Configuration Wizard** dialog box appears.
5. Click **Yes** to proceed with the Configuration Wizard or click **No** if you plan to configure your project later.

For more information on how to configure Issue Manager, see *Issue Manager Administration*.

Issue Manager Administration

In addition to the management of user accounts and other administration tasks, Issue Manager also shares its database repository with Silk Central. Issue Manager projects are also created, configured, and managed in the Silk Central **Administration** unit. This enables complete integration of Issue Manager projects with Silk Central projects.

When using Issue Manager in conjunction with Silk Central, projects may contain numerous elements in addition to issues, including requirements, tests, and execution definitions.

Issue Manager projects

Issue Manager uses a project-based workflow. Only those issues that are associated with the active project can be viewed or processed. You must select an Issue Manager project (via the **Projects** unit) before you can gain access to individual issues. The selected project remains the default selected project each time you access Issue Manager, until you select an alternate project to work with, at which point the newly selected project becomes the default.

Initializing database configuration

Once a project has been created, its database connection must be initialized before it can be used as an Issue Manager project.

Setting Up Your Data

Once you feel comfortable using Issue Manager, you can enter information about your organization into a database. You can enter this information in the **Configuration** page. To open the **Configuration** page, click **Issues > Configuration**.

Your first step should be to gather information about your organization's work processes. Take the time to plan carefully. To effectively use Issue Manager in your organization you have to gather information about the following major areas:

- People.
- Products.
- The suitability of the **Issue Tracking** page, the action dialog boxes, and the workflow of Issue Manager for your organization.

Complete the following activities in the following order:

1. Gather information about the people in your organization.
2. Gather information about products.
3. *Optional:* Assess the interface and the workflow of Issue Manager. You can customize the **Issue Tracking** page, the action dialog boxes, the severities list, and the workflow in the **Configuration** page.
4. Create your own database and populate it with the setup data.

Gathering Information About the People in Your Organization

To gather information about the people in your organization:

1. Draw up a list of groups.
2. Decide which security privileges and initial issue states are appropriate for all members in each group.
3. Draw up a list of accounts.
Each user who logs into Issue Manager is required to have an account.
4. Decide on a naming scheme for your accounts.
Include the assigned group, and, if necessary, one or more Inboxes. Ask yourself the following questions:
 - Which accounts need extra security privileges that are not covered by the group's privileges?
 - Should the initial issue states for this account differ from those of the group?
 - Do you want login to require passwords?
5. While reviewing the list of user accounts, note which users need Inboxes.
Do groups need Inboxes, too? Decide on an inbox naming scheme.
6. You may customize the default view for all users.
All users have the same default view of Inboxes. The view contains eight columns of information.

Gathering Information About Products

To gather information about the products in your organization:

1. Draw up a list of products, releases for each product, and platforms for each release.
2. Compare your platform list with the list in **Administration > Platforms**.
3. Draw up a list of software and documentation components of each product.
Components (functional areas) are not required; however, they allow you to track issues based on product, release, platform, and component, and to gain finer control over the routing of issues.
4. Draw up a list of issue severities.
The values on this list are displayed in the **Severity** column on the **Issue Tracking** page. You can categorize this list by product or issue type, or you can come up with a generic list for all products and issue types.
5. Compare your list with the list in **Administration > List of Values**.
6. Decide which inbox should receive an issue pertaining to a given product, component, release, and platform.
These are *routing rules*.

Optional: Assessing the Interface and the Workflows

To assess the interface and the workflows of Issue Manager:

1. Assess the appearance and behavior of the **Issue Tracking** page to determine whether or not you want to perform the following tasks:
 - Modify the default column labels.
 - Add columns to capture additional information.

- Modify the fields on the **Action** dialog box, or add your own fields.

Keep a list of your modifications. It's your responsibility to inform the users in your organization about any modifications that you make.

2. Assess the default workflows to determine whether or not they model your organization's needs.

Look at each state and the valid actions for that state. Are the default group permissions for that action satisfactory?

Routine Administrative Tasks

Occasionally, after initial setup is complete, you will have routine administrative tasks to perform to keep Issue Manager up to date. The following are the most common tasks:

Adding a new user

For a new user, you will need to perform the following tasks:

- Create an Inbox, if needed, before setting up the account.
- Set up a new account, associating the account with an existing group and, if necessary, a default inbox.
- Add the user's Inboxes to the routing rules if the user needs to receive issues through automatic routing.

Removing a user

When a user leaves your organization, you should perform the following tasks:

- Immediately deactivate the user's account so that he or she can no longer log in. You cannot delete an account if a user has created or modified issues.
- Replace any references to the user's Inbox in the routing rules with another Inbox. This step prevents new issues from being sent to this Inbox.



Note: You should make changes to routing rules only when no other users are logged in to Issue Manager.

- Clear out the user's Inbox by reassigning the issues to other Inboxes.
- Delete the user's Inbox.

Shifting work assignments

You will need to update group assignments when users change jobs. For example, say that Sarah, a QA engineer, moves to development. You will also need to modify the routing rules so that users will receive issues that are appropriate for their new responsibilities. For example, Sarah should now receive issues that require attention by development instead of QA.

When users change projects, you need to modify the routing rules to reflect their current responsibilities. You also need to make sure that the default inbox is still appropriate. For example, say that Judy is a writer who used to write manuals for Product A, but now documents Product B. Judy's default inbox, Judy - Doc (Product A), should be changed to Judy - Doc (Product B).

Adding new product information

Sometimes you will need to perform the following tasks:

- Add new products.
- Add additional releases, platforms, and components for existing products.
- Change your routing rules to accommodate new products, components, releases, and platforms.

Optional Features

Once you have set up your database and have become comfortable in using Issue Manager, consider using the following optional features:

Email notification

The email notification feature allows Issue Manager users to have email sent to them whenever certain events occur to an issue, for example a reassignment or a state change. Users with the **Manage email notifications** security privilege can define additional circumstances under which an email is sent. Consult your users about the circumstances under which they would like to receive email.

Archiving issues

The greater the number of issues in the database, the longer actions, for example queries, may take to execute. To help you improve the overall performance, Issue Manager allows you to archive issues that are no longer relevant to your organization's efforts. Archiving issues segregates them in the database so that you can, for example, run queries against only the active issues.

Setting Up Groups

Depending on your organization, groups can be entire departments, cross-departmental project teams, or intra-departmental teams.

Each group can have distinct security privileges and permissions. An Issue Manager group also determines where issues reported by the members of the group enter the workflow. A user's group determines what the user can do in Issue Manager and affects how issues reported by the user are routed.

The sample database provides the following default groups:

- Corporate
- Development
- Documentation
- Quality Assurance
- R&D Management
- Technical Support

All Issue Manager users, regardless of the tasks that they perform, must be assigned to a group. A user can only be in one group at a time.

Groups must be set up before inboxes and user accounts are set up, because each inbox and user must be associated with a group.

 **Note:** Groups are created, edited, and deleted in the **Groups** tab in **Administration > User Management**, while group settings specific for Issue Manager are configured in **Issues > Configuration > Groups**.

Group Properties

Groups have a number of properties. Basic group information, for example the name of a group, is specified in the **Groups** tab in **Administration > User Management**. Group settings that are specific for Issue Manager, for example the initial issue states and certain security privileges, are specified in the **Groups** tab in **Issues > Configuration**.

 **Note:** You have to create a group before you can modify group settings that are specific for Issue Manager.

Group Information

The following table lists the properties that you can set for a group in **Administration > User Management**:

| Property | Description |
|----------|---|
| Name | Required unique name for a group, for example <i>Documentation</i> . Can contain up to 20 characters. |

| Property | Description |
|-----------------------------|--|
| Description | <i>Optional:</i> Description of the group, for example <i>Documentation produces all technical publications, including manuals and release notes.</i> Can contain up to 80 characters. |
| Account and Role Assignment | Users must have certain roles assigned to them. |
| Project Assignment | A group must be assigned to at least one project. Users assigned to a group have access only to the projects that are assigned to the respective group. |

Initial Issue State

The *initial issue state* is the first state assigned to a bug, enhancement, or documentation issue when it is submitted by a member of a given group. Different groups can have different initial states for the same type of issue. Moreover, within a group each issue type, bug, enhancement, and documentation issue, can have a different initial state.

For example, you might assign a software bug submitted by members of the Development group to an initial state of *Dev-Ready* because you assume that developers can accurately assess what is actually a software bug. On the other hand, you might assign a software bug submitted by members of the Corporate group to an initial state of *Unreviewed* because you might want to first verify that the issue is truly a new bug before sending it on to Development to be fixed.

Issue Manager provides a number of default states for you to choose from. You can also create your own issue states before you set up the groups in your organization.

Although initial issue states apply to every member of a group, you can explicitly override these settings for a given user.

Group Security Settings

Security privileges define which activities all users in a group can perform. You can assign additional privileges for a given user in a group by editing his or her user account but you cannot take away privileges that have been granted to the user's group.

To assign security privileges for a group, click **Define Group Settings** in **Issues > Configuration > Groups**. Check the check box to assign the corresponding privilege to the group. To give a group superuser privileges, check all check boxes.

The following privileges allow group members to perform specific customization tasks:

| Check Box | Description |
|---|---|
| Manage groups, inboxes and user accounts | Create and edit groups, inboxes, and user accounts, including security privileges. |
| Manage products | Create and edit products, releases, platforms, and components. Edit the Platforms list and other lists. |
| Manage routing rules | Define how an issue moves from one inbox to another based on the product, release, platform, and component it was logged against. |
| Manage GUI customization | Customize fields and tabs on the Issue Tracking page. |
| Manage workflow customization | Define issue states and actions, action permissions, help text, and the appearance and usage of action dialog boxes. |
| Manage email notifications | Define email notification rules and add email notifications that apply to all issues in the database. |

| Check Box | Description |
|-----------------------|--|
| Archive issues | Removes selected issues from the issues and the tables related to the issues and places them in the ARCHIVE and the tables related to the ARCHIVE. |

The following table lists the privileges that affect the routing of issues and email notifications:

| Check Box | Description |
|---------------------------------------|---|
| Reassign issues | Move an issue from one inbox to another. |
| Issue verification preferences | Check to override the normal routing rules. |
| Email notification per issue | Add email notifications to individual issues. |
| Enter/edit issues | Permission to enter and edit issues. |

Setting Up Inboxes

An inbox is similar to a mail box. All open, active issues addressed (assigned) to a particular user or group of users are delivered to the Inbox. It is important not to confuse an Inbox with a user account. Issues are always assigned to inboxes, not to specific users.

 **Note:** Archived issues do not display in inboxes.

How are issues routed to inboxes?

The routing rules of Issue Manager automatically dispatch issues to the appropriate inboxes. An issue stays in an inbox until a user acts on it, for example when a developer fixes a bug. Then Issue Manager automatically dispatches the issue to the next inbox, based on your workflow and routing rules. Finally, when an issue is closed, Issue Manager removes the issue from the last inbox.

An issue can also be routed to another inbox when a user with the **Reassign issues** security privilege manually reassigns it to another inbox.

Which users need inboxes?

Users who take action on issues should have an inbox. Developers, QA engineers, and technical writers are in this category. Users who only report or view bugs do not require inboxes. Technical support personnel who routinely report bugs found by customers, but who do not act on bugs may be in this category.

How many inboxes does a user need?

A user can have multiple inboxes. For example, a developer might want two separate inboxes to hold bug reports from two different products. In this case, you might set up the two inboxes *Dan - Dev (Product A)* and *Dan - Dev (Product B)*.

Individual and group inboxes

You can set up inboxes for individual users, groups of users, or subsets of issues. For example, the QA engineers Sarah and Mike might be responsible for verifying bug fixes for different products. In such an instance you would set up the two inboxes *Sarah - QA* and *Mike - QA*.

Users can also share inboxes. For example, a Product Management group might need only one inbox into which all enhancements are funneled; periodically, the entire group might meet to review the suggested improvements. In such situation you might set up a single inbox called *PM - Inbox*.

Another use of an inbox might be as a holding place for a subset of issues. For example, you might want to create an inbox for all deferred issues for Product C. In such a case, creating an inbox called *Deferred (Product C)* might be appropriate.

Assigning a user's default inbox

When you set up a user account you will assign an inbox to the user account. This inbox is the default inbox of the user. A user can have only one default inbox.

Viewing default and other inboxes

You can view your default inbox by selecting **Issues > Issue Tracking**. If the default inbox is not displayed, click **My Inbox** in the toolbar.

To see an inbox other than the default, select another inbox listed in the **Inbox** list box. Any user can view any inbox, but only a user with the correct privileges and permissions can act on or reassign an issue.

System default view

Issue Manager provides an initial default view of inboxes for all users. You can customize the default view of inboxes for all users.

Adding an Inbox

You should add at least one inbox for every user who acts on issues.



Note: You need the **Groups, inboxes and user accounts** security privilege to add, edit, or delete inboxes.

To add an Inbox:

1. In the menu, click **Issues > Configuration**.
2. Click **Inboxes**. The **Inboxes** page opens.
3. Click **Add Inbox**. The **Inbox Settings** dialog box opens.
4. Type a name for the Inbox in the **Name** field.

Each inbox requires a unique name of up to 50 characters. The following are sample Inbox names:

- Dave (Dev)
- PM Inbox
- Jesse - Doc
- Dan - Dev (Product A)
- Dan - Dev (Product B)

User Inbox names can be in any format you choose, but you should follow the format consistently. The following are sample formats:

- user (group)
- user - group
- user - group (Product-name)
- user

Group inboxes should also follow a consistent format, for example one of the following:

- group
- Group (Product-Name)

Special characters are not allowed. Inbox names are displayed in ASCII sort order (capital letters first) on the **Inboxes** page.

5. From the **Group** list box, select the group that is associated with this Inbox. Based on this association, Issue Manager helps users reassign issues selectively. For example, say that Denise, who is in the Documentation group, reassigns an issue in her inbox. Issue Manager opens the **Issue Reassignment** dialog box with Documentation as the default group. Denise can then select another member of the Documentation group from the **New Inbox** list box without having to scan the entire list of inboxes.

6. *Optional:* Enter a description of the inbox in the **Description** field.

The description can have a length of up to 80 characters. For example `Developer David Hanson's inbox` or `dhart's inbox for Product B issues`.

7. The **Is Active** check box is checked by default.



Note: If the inbox is used in the routing rules you cannot uncheck the **Is active** check box.

8. Click **OK** to create the new inbox..

Editing an Inbox



Note: You need the **Groups, inboxes and user accounts** security privilege to add, edit, or delete inboxes.

To edit an inbox:

1. In the menu, click **Issues > Configuration**.
2. Click **Inboxes**. The **Inboxes** page opens.
3. Click on the name of the inbox that you want to edit. The **Inbox Settings** dialog box opens.
4. Edit the name of the inbox.
5. Select a different group from the **Group** list box.
6. Edit the description of the inbox.
7. Check or uncheck the **Is active** check box.



Note: If the inbox is used in the routing rules you cannot uncheck the **Is active** check box.

8. Click **OK** to save your changes.

Deleting an Inbox

You can delete an inbox only if it does not contain any issues, is not referenced in routing rules, and is not the default inbox of a user. Before you try to delete an inbox, edit the routing rules to prevent new issues from being routed to the inbox, and then reassign the existing issues in the inbox.



Note: You need the **Groups, inboxes and user accounts** security privilege to add, edit, or delete inboxes.

To delete an inbox:

1. In the menu, click **Issues > Configuration**.
2. Click **Inboxes**. The **Inboxes** page opens.
3. In the **Actions** column of the inbox that you want to delete, click **X**. A confirmation dialog box opens.
4. Click **Yes** to remove the inbox from the system.

Setting Up the Initial Default Inbox View

When you install Issue Manager, the initial default inbox view consists of the following eight columns:

- Severity
- Issue#
- Product
- Component
- Release
- Synopsis
- Created By
- Changed By

You can perform the following customization tasks to the initial default inbox view for all users:

- Add or remove columns from the display.
- Edit the label of a column.
- Change the sort order of a column.

Individual users can create custom initial views of inboxes.

Modifying the Default Inbox View



Note: You need the **Groups, inboxes and user accounts** security privilege to add, edit, or delete inboxes.

To modify the initial default inbox view:

1. In the menu, click **Issues > Configuration**.
2. Click **Inbox Views**. The **Inbox Views** page opens. This page displays all of the fields that you can choose to display in an inbox view. Most of the fields are recognizable from the **Issue Details** page. Others, for example **NumHistory**, **NumCases**, **NumAttaches**, **NumTestcases**, and **NumNotify** refer to the number of entries on the respective **History**, **Customers**, **Attachments**, **Testcases**, and **Notifications** pages of the **Issue Details** page. The **Inbox Views** page also displays the schema, or **DB Field Name**, of each column.
3. From the **View of user** list box, select **SYSTEM DEFAULT**.
To update the inbox view of other users, select the respective user instead.
4. To add a column to the inbox, click the ➔ that corresponds to the column in the **List of Available Fields**.
5. To remove a column from the inbox, click the ⬅ that corresponds to the column in the list of the inbox columns.
6. To change the label of a column and the sort order, perform the following steps:
 - a) Click on the name of the inbox column in the **Inbox Column Label** column. The **Inbox Column Properties** dialog box opens.
 - b) Type a new label for the column into the **Inbox column label** field.
 - c) Click the appropriate option button to change the sort order.
 - d) Click **OK**.
7. To move a column to the left in the inbox, click ⬅. To move a column to the right in the inbox, click ➔.
8. Click **View Inbox** to return to the **Issue Tracking** page.



Note: If you updated the system default inbox view or the view of another user, the changes are applied when the respective users log out and back in again.

Setting Up User Accounts

A user account is a collection of information about a given user. Each user is assigned to a group and, if the user acts on issues, an inbox. A user account can be password-protected. Every person who logs on to the system to report an issue, act on an issue, or merely browse, needs a user account.



Note: You can create, edit, or delete user accounts in the **Accounts** tab in **Administration > User Management**.

Users inherit the security privileges and the initial issue states of their group. To give an individual user enhanced privileges or alternate initial issue states, you must edit the account of this user. Privileges defined by a group setting cannot be removed. A SuperUser, an Administrator, or a Project Manager always has all privileges, independent of whether the user is a member of a group with fewer privileges. To configure these settings, use the **User Accounts** tab in **Issues > Configuration**.



Note: Before you can set up user accounts, you must have already set up groups and inboxes.

User Account Properties

The issue-related properties for user accounts are specified in the **User Accounts** tab in **Issues > Configuration**.

 **Note:** Before you can view or edit issue-related account properties, you must create a user account.

Click on the name of a user in the **User Accounts** tab to edit the issue-related account properties.

The following table lists the issue-related account properties:

| Property | Description |
|---------------|--|
| Login | Required unique login name for a user. Can contain up to 30 characters. Special characters are not allowed.  Note: You cannot edit the login name in Issue Manager. |
| Default Inbox | The default inbox of the user. You can select an alternate default inbox for the user from the Inbox list box. |
| Setting Group | Required. The group to which this user belongs. When you select a group from the Setting Group list box, Issue Manager automatically assigns the default values of the group for initial issue states and group security settings to the user. |

Initial Issue State

The *initial issue state* is the first state assigned to a bug, enhancement, or documentation issue when it is submitted by a user. The default values are the group defaults, which are the values assigned to the group to which the user belongs.

You might want to change the default initial issue states for a user account, if, for example, a user has a higher level of technical competence. For example, suppose that within a group of non-technical managers, one manager is decidedly technical. You might want to have this user's issues enter Issue Manager further along in the workflow.

User Security Settings

Security privileges define which activities the user can perform. The default security privileges for a user are the security privileges of the group to which the user belongs. You can assign additional privileges for a given user in a group by editing his or her user account but you cannot take away privileges that have been granted to the user's group. The default security privileges are read-only.

To assign security privileges to a user account, click on the name of the user in **Issues > Configuration > User Accounts**. Check the appropriate check box to assign the corresponding privilege to the user.

The following table lists the privileges that allow users to perform specific customization tasks:

| Check Box | Description |
|---|---|
| Manage groups, inboxes and user accounts | Create and edit groups, inboxes, and user accounts, including security privileges. |
| Manage products | Create and edit products, releases, platforms, and components. Edit the Platforms list and other lists. |
| Manage routing rules | Define how an issue moves from one inbox to another based on the product, release, platform, and component it was logged against. |
| Manage GUI customization | Customize fields and tabs on the Issue Tracking page. |

| Check Box | Description |
|--------------------------------------|--|
| Manage workflow customization | Define issue states and actions, action permissions, help text, and the appearance and usage of action dialog boxes. |
| Manage email notifications | Define email notification rules and add email notifications that apply to all issues in the database. |
| Archive issues | Removes selected issues from the issues and the tables related to the issues and places them in the ARCHIVE and the tables related to the ARCHIVE. |

The following table lists the privileges that affect the routing of issues and email notifications:

| Check Box | Description |
|---------------------------------------|---|
| Reassign issues | Move an issue from one inbox to another. |
| Issue verification preferences | Check to override the normal routing rules. |
| Email notification per issue | Add email notifications to individual issues. |
| Enter/edit issues | Permission to enter and edit issues. |

Editing a User Account

 **Note:** You need the **Groups, inboxes and user accounts** security privilege to edit user accounts.

To edit the security privileges and the initial issue states for a user account:

1. In the menu, click **Issues > Configuration**.
2. Click **User Accounts**. The **User Accounts** page opens.
3. Click on the name of the user for whom you want to edit the security privileges. The **User Settings** dialog box opens.
4. Change the security privileges of the user and the initial issue states.
5. Click **OK** to save your changes or click **Cancel** to exit without making changes.

Defining Product Settings

In addition to tracking issues for specific products, Issue Manager also tracks issues for the following:

- Specific versions of products.
- Specific platforms that product versions are associated with.
- Specific product components.

Product teams often have more than one person writing code for a given application. Issue Manager enables you to link products and components with specific inboxes. Additionally, Issue Manager automatically assigns new issues to the appropriate developers based on the product, the component, the release and the platform.

 **Note:** You can create and edit products, components, versions, builds, and platforms in the **Administration** unit of Silk Central. The only properties that you can configure through Issue Manager are product inbox settings.

It is up to you to define how you want to distinguish products from components. A product in one organization might be defined as a component in another organization. The following basic definitions might help you:

- Product** A product is a piece of software that can be run separately from other software that may be delivered with it. It can be sold separately if required. In a product the executable program is separated from the code base. A product may consist of multiple components.
- Component** A component is a functional area of a product. A component cannot be sold separately, and users do not see a component as a distinct product. A product may have one or more components associated with it.
- Release** A release is a specific version or a specific software build of a specific product. A release can be an internal development build or a build that is shipped to customers. A product might have multiple releases associated with it. A specific release of a product might be available on more than one platform.
- Platform** A platform is an environment in which a product runs. A platform consists of a combination of any of the following elements:
- Hardware.
 - Operating system, for example Microsoft Windows.
 - Operating system release, for example 8.1.

Many of the values that are defined for products, releases, platforms, and components are displayed on the **Products** page in **Issues > Configuration > Products**.

Defining Inbox Settings for a Project

To define the inbox settings for specific products in Issue Manager:

1. In the menu, click **Issues > Configuration**.
 2. Click **Products**. The **Products** page opens.
 3. Click **Define Product Settings**. The **Product Settings** dialog box opens.
 4. From the **Product** list box, select the product for which you want to define inbox settings.
-  **Note:** By clicking the name of a product that has already been assigned to an issue in the **Products** page, you can bypass this step.
5. From the **QA-Inbox** list box, select the predefined inbox to which an issue assigned to QA and defined with the given product should be routed.
 6. From the **Dev-Inbox** list box, select the predefined inbox to which an issue assigned to development and defined with the given product should be routed.
 7. From the **Enh-Inbox** list box, select the predefined inbox to which an enhancement request issue which is defined with the given product should be routed.
 8. From the **Doc-Inbox** list box, select the predefined inbox to which an issue assigned to documentation and defined with the current product should be routed.
 9. Click **OK**.

Deleting a Product

Deleting a product in Issue Manager does not actually delete the product from the repository, but rather detaches the product from the selected project, so that the product is no longer accessible from Issue Manager.

 **Note:** You can only delete a product as long as no issues have been entered for the specific product.

To delete a product from the active project:

1. In the menu, click **Issues > Configuration**.
2. Click **Products**. The **Products** page opens.
3. In the **Actions** column of the product that you want to delete, click **X**. A confirmation dialog box opens.

4. Click **Yes** to complete the deletion.

Setting Up Routing Rules

Issue Manager relies on defined routing rules, the current state of each issue, and the corresponding state owners to determine the inboxes that issues are to be routed to during their life cycles. This sophisticated routing mechanism replaces what would otherwise be a tedious task for issue dispatchers.

The state of an issue is the current condition of the issue. A number of states are provided in the default workflow of Issue Manager. The state owner is the role in your organization that is responsible for acting on an issue in a given state.

The following are examples of issue states and owners from the default issue workflow:

Unreviewed Someone needs to determine whether or not this issue is truly a bug or documentation error. Usually, this role is owned by a QA engineer.

Dev-Ready Code is ready to be addressed. This role is typically owned by a developer.

QA-Ready Someone needs to verify that the bug has actually been fixed. This role is usually owned by a QA engineer.

Different issues of the same type can enter the workflow in different states, depending on who submits them. The routing is affected accordingly. For example, an issue submitted by a developer enters the workflow in the Dev-Ready state, and is therefore routed to a developer's inbox. An issue submitted by a corporate user enters the workflow in the Unreviewed state, and is therefore routed to a QA engineer's inbox. The so-called initial issue state is assigned through group settings.

An issue moves from one state to the next in the workflow when a user takes action on that issue. For example, when a QA engineer confirms that a reported, unreviewed issue is indeed a bug, the issue moves from the Unreviewed state to the Dev-Ready state. The main goal of using routing logic here is to make sure that once the issue is judged to be a bug by the QA engineer, it moves from his or her inbox to the appropriate developer's inbox-without need for manual intervention.

Routing Rules

Issue routing is based on rules that you define for a product and its associated releases, platforms, and components. This rule-based mechanism gives you fine control over the distribution of issues because issues are routed based in part on a combination of the following criteria:

- Product
- Component
- Release
- Platform
- Issue state
- State owner

You need to assign the following four inboxes for each set of criteria:

- An inbox responsible for verifying issues, which is typically a QA inbox.
- An inbox responsible for fixing issues, which is typically a Development inbox.
- An inbox for handling documentation problems, which is a Documentation inbox.
- An inbox for enhancement requests, which is typically a product management inbox.

Example

All *Product A* bugs related to the *Installer* component and associated with the *Motif* platform can be routed to one set of inboxes, while all *Product A* bugs related to the *Installer* component for the *Windows* platforms can be sent to another set of inboxes.

Default routing

In addition to granular routing rules, Issue Manager requires that you define one routing rule for the product as a whole, which means default routing. In default routing, issues for all components, releases, and platforms of a specific product are routed to a designated set of four inboxes.

Each inbox covers one of the following areas:

- QA
- Development
- Documentation
- Enhancement requests

For example, all requests for enhancements for *Product X* are directed to the one enhancement request inbox you assign, regardless of the individual component, platform, or release.

Issue Manager uses the default routing rule only when other routing rules do not exist or are not applicable. In other words, default routing rules are applied only when specific rules do not match or have not been specified.

Default routing is set up during Issue Manager product setup.

Analyzing the processes in your organization

Analyze the breakdown of work in your organization. Review your list of products, releases, and platforms and consider each component in turn against this list. Ask yourself questions such as the following:

For this component in this release and on this platform, who, which means which inbox, is responsible for each action, for example verifying, fixing, and so on?

Then define as many rules as required for the different combinations of the following four criteria:

- Product
- Release
- Platform
- Component

Specifying Rules

Each rule is entered into the **Routing Rules** page, which is located in **Issues > Configuration > Routing Rules**.



Note: The **New Routing Rule** dialog box has an if-then layout. If an issue matches specific conditions, which means it pertains to a particular release, platform, component, and issue type, then it is routed to one of the four specified inboxes. Which one of the four inboxes is chosen is determined by the issue's current state and owner.

The wildcard % symbol

The percent sign (%) in the **Release**, **Platform**, and **Component** fields of the **New Routing Rule** dialog box serves as a wildcard character that matches all characters. Using the percent sign by itself in all three fields would be the same as default routing; everything would be routed to the four inboxes specified in the dialog box, regardless of values for release, platform, and component.



Tip: You must specify release, component, and platform names consistently to take advantage of wildcarding.

Once a rule has been saved, it is entered into the **Routing Rules** page. This page is essentially a routing table for a particular product. Alternate product selections can be made using the **Product** list box at the top of the page.

The order of rules is critical

The order of rules for a component is critical. Issue Manager routes issues by evaluating their current state, for example Unreviewed, against each rule in the table in order. As soon as Issue Manager finds a match, it executes the rule. If there is no match, it routes the issue according to the default rules. Default routing rules are applied only after all other rules for a given product have been applied.

Example

Release 4.% in the **Release** field would match all Release 4 releases: 4.0, 4.1, 4.1.1, and so on.

Example: How to read a routing rule

For example, send issues that are related to *Product A* for all releases and all platforms to the *Sonja - Dev* inbox when the issue is ready for development, to the *Mike - QA* inbox when the issue is ready for QA, to the *Product A request-for-enhancement* inbox *Dan -- Dev (Product A)* when an enhancement is submitted, and to the *Judy -Doc* inbox when a documentation issue relating to the *Show me* component is reported.

Adding Routing Rules

To add a routing rule:

1. In the menu, click **Issues > Configuration**.
2. Click **Products**. The **Products** page opens.
3. From the **Product** list box, select the product for which you want to configure a routing rule.
4. Click **New Routing Rule**. The **New Routing Rule** dialog box opens.
5. Click the corresponding option button to define if the issue is a **BUG or ENHANCEMENT (SOFT)** or if the issue is a **DOC-ISSUE (DOC)**.
6. Select a predefined release from the **Release** list box. Or type in a predefined release into the **Specific Matchcode** field. Select nothing, which means the dotted line, if any release should be considered.
7. Select a predefined platform from the **Platform** list box. Or type in a predefined platform into the **Specific Matchcode** field. Select nothing, which means the dotted line, if any release should be considered.
8. Select a predefined component from the **Component** list box. Or type in a predefined component into the **Specific Matchcode** field. Select nothing, which means the dotted line, if any release should be considered.



Note: You must select a value for at least one of either release, platform, or component. Otherwise, the new routing rule would follow the same rules as the default rule.

9. In the **Then route to these inboxes** area of the dialog box, select a predefined inbox for each of the following four inbox categories:
 - **QA Inbox**
 - **Development Inbox**
 - **Enhancement Inbox**
 - **Documentation Inbox**
10. Click **OK**. The rule is saved and added to the routing table.



Note: If you have made changes to the routing table, Issue Manager saves and reconfigures the routing rules, which may take some time.

Editing Routing Rules

To edit an existing routing rule:

1. In the menu, click **Issues > Configuration**.
2. Click **Products**. The **Products** page opens.
3. From the **Product** list box, select the product for which you want to configure a routing rule.
4. In the **Actions** column of the rule that you want to edit, click . The **Edit Routing Rule** dialog box opens.
5. Click the corresponding option button to define if the issue is a **BUG or ENHANCEMENT (SOFT)** or if the issue is a **DOC-ISSUE (DOC)**.
6. Select a predefined release from the **Release** list box. Or type in a predefined release into the **Specific Matchcode** field. Select nothing, which means the dotted line, if any release should be considered.
7. Select a predefined platform from the **Platform** list box. Or type in a predefined platform into the **Specific Matchcode** field. Select nothing, which means the dotted line, if any release should be considered.
8. Select a predefined component from the **Component** list box. Or type in a predefined component into the **Specific Matchcode** field. Select nothing, which means the dotted line, if any release should be considered.



Note: You must select a value for at least one of either release, platform, or component. Otherwise, the new routing rule would follow the same rules as the default rule.

9. In the **Then route to these inboxes** area of the dialog box, select a predefined inbox for each of the following four inbox categories:

- **QA Inbox**
- **Development Inbox**
- **Enhancement Inbox**
- **Documentation Inbox**

10. Click **OK**. The rule is saved and added to the routing table.



Note: If you have made changes to the routing table, Issue Manager saves and reconfigures the routing rules, which may take some time.

Deleting Routing Rules

To delete an existing routing rule:

1. In the menu, click **Issues > Configuration**.
2. Click **Products**. The **Products** page opens.
3. From the **Product** list box, select the product for which you want to configure a routing rule.
4. In the **Actions** column of the rule that you want to delete, click . A **Delete Routing Rule** confirmation dialog box opens.
5. Click **Yes** to complete the deletion.

Reordering Routing Rules

To reorder routing rules:

1. In the menu, click **Issues > Configuration**.
2. Click **Products**. The **Products** page opens.
3. From the **Product** list box, select the product for which you want to configure a routing rule.
4. In the **Actions** column of the rule that you want to reorder, click one of the following:
 - To move the rule up in the list, click .
 - To move the rule down in the list, click .

Automatic Routing Override Preferences

You can allow individual users or entire groups to override the normal routing for verifying issues in your organization. For example, a user might want to verify that the reported issues have been fixed.

To permit such an override, assign the **Issue verification preferences** privilege to the user or group. With this security privilege, users can then choose one of the following three strategies from **Issues > Configuration > Preferences**:

| Strategy | Description |
|-------------------------------------|---|
| Always use normal routing | Normal routing for verification. The issue will be routed to the appropriate QA inbox. |
| Always verify your own issue | Have Issue Manager route issues that users submit to their own inboxes for verification, rather than to the typical QA inbox. |
| Prompt for each new issue | Have Issue Manager give users the option to override normal routing each time an issue is reported. |

Reassignment overrides automatic routing

Reassigning an issue is a method of manual routing that overrides the automatic routing mechanism. A user can explicitly take the Reassign action to route an issue to the inbox of another user, typically a user in the same group. For example, developer Bill, who is going on vacation, might reassign a Dev-Ready bug to his developer colleague Dan. The bug remains in the Dev-Ready state, but it has been reassigned.

When an issue that has been reassigned returns to a previous state in the workflow, for example if Dan were to reject a bug fix, Issue Manager remembers the reassignment and later returns the issue to the inbox to which the issue was manually routed, which in this case is Dan's inbox, rather than to the inbox to which it was originally routed, which in this case is Bill's inbox.

Example

To appreciate how efficiently reassignment works, consider an example in which a bug is reassigned twice - once in the Dev-Ready state and once in the QA-Ready state, and the QA engineer rejects a bug fix, sending the bug back to an earlier state in the workflow.

1. A new bug enters the workflow in the *Dev-Ready* state. The routing rules automatically send the bug to the inbox of developer Bill.
2. Bill reassigns the bug to the inbox of developer Dan.
3. Dan claims that the bug has been fixed, which sends the bug to the *QA-Ready* state.
4. The routing rules automatically route the bug to the inbox of QA engineer Mike for verification.
5. Mike reassigns the bug to the inbox of his colleague Sarah.
6. Sarah rejects the fix, which sends the bug back to Dev-Ready.
7. Issue Manager routes the Dev-Ready bug back to the inbox of Dan, because he was the last developer to act on the bug in that state.
8. Dan fixes the bug again, sending the bug back to QA-Ready.
9. Issue Manager routes the QA-Ready bug back to Sarah's inbox, because she was the last QA engineer to act on the bug in that state.

Without enhanced routing, the issue in the Dev-Ready state (step 7) would be routed to Bill's inbox, who would then once again reassign the issue to Dan. Similarly, the issue in the QA-Ready state (step 9) would be routed to Mike's inbox, who would then once again reassign the issue to Sarah.

Customizing the Issue Details Page

Issue Manager allows you to change the appearance and behavior of the **New Issue** page and the **Issue Details** page for an existing issue. For example, you might want to make a field label more relevant for your organization, for example Build rather than Release.

Changes you make to the labels on the **Issue Details** page are propagated to most other pages and dialog boxes that are invoked through the **Customization** of Issue Manager.

Issue Manager allows you to make the following changes to the **Issue Details** page:

- Edit the labels and modify the properties of the fields that specify basic information about an issue. For example, you might want to change the label Synopsis to Summary.
- Edit the labels of the fields in the **Automatic Fields** area.
- Edit the tab labels.
- Define fields on Custom Tab 1 and Custom Tab 2. You can modify or delete these fields or add your own. By default, Custom Tab 1 contains four fields.

The following table describes where to look for information related to customizing a standard issue field label and standard issue field contents:

| Name | Edit field label? | Customize field properties? |
|-----------------------|--|---|
| Issue Type | <i>Customizing Fields and Tab Labels</i> | Yes. |
| Product | <i>Customizing Fields and Tab Labels</i> | Yes. See <i>Editing Products</i> . |
| Release | <i>Customizing Fields and Tab Labels</i> | Yes. See <i>Editing Versions</i> . |
| Platform | <i>Customizing Fields and Tab Labels</i> | Yes. See <i>Editing Platforms</i> . |
| Component | <i>Customizing Fields and Tab Labels</i> | Yes. See <i>Editing Components</i> . |
| Severity | <i>Customizing Fields and Tab Labels</i> | Yes. |
| Synopsis | <i>Customizing Fields and Tab Labels</i> | No. Users fill in the synopsis. |
| Issue Number | <i>Customizing Fields and Tab Labels</i> | No. The issue number cannot be customized. |
| Assigned To | <i>Customizing Fields and Tab Labels</i> | Yes. See <i>Routing Rules</i> . |
| State | <i>Customizing Fields and Tab Labels</i> | Yes. See <i>Customizing Workflow States</i> . |
| Reason Code | <i>Customizing Fields and Tab Labels</i> | Yes. See <i>Reason Code</i> . |
| Action Release | <i>Action Properties</i> | Yes. |

The following table describes where to look for information related to customizing a tab label and fields on tabs:

| Name | Edit tab label? | Customize field on tabs? |
|--------------------|--|---|
| Description | <i>Customizing Fields and Tab Labels</i> | No. Users fill in the Description tab. |
| History | <i>Customizing Fields and Tab Labels</i> | No. Issue Manager fills in the History tab, but you can customize the Action column. See <i>Current State Information</i> . |
| Customers | <i>Customizing Fields and Tab Labels</i> | No. Users can take the predefined Add Customer action. |
| Custom Tab | <i>Creating Custom Tabs</i> | Yes. See <i>Editing Custom Fields</i> . |

Customizing Fields and Tab Labels

 **Note:** You need the `GUI customization` privilege to perform the actions described in this topic.

This topic explains how to customize the standard issue fields, tab labels, and automatic fields, with the exception of the **Action Release** field. For information on customizing the **Action Release** field, see *Standard Action Fields Tab*. You cannot delete any of these fields, nor can you add to them. However, you can change their properties. For example, you can edit their labels, choose their mode, and restrict the groups that have permission to fill in or modify them.

To modify the standard issue fields and tab labels:

1. In the menu, click **Issues > Configuration**.
2. Click **Standard Issue Fields**. The **Standard Issue Fields** page opens.

 **Note:** The **Standard Issue Fields** page is laid out like the **Issue Details** page.

3. If you want to rename a tab label, perform the following actions:
 - a) Click the link of the tab that you want to edit. The **Edit Label** dialog box opens.
 - b) Modify the name of the tab as required.
 - c) Click **OK**.
4. If you want to edit the properties of a field, perform the following actions:
 - a) Click the link of any field that you want to edit. The **Edit Field Properties** dialog box opens.
 - b) Modify the properties of the field as required.
 - c) Click **OK**.

GUI Customization Properties

The fields on the **Issue Details** page have properties that determine their appearance and usage. For example, the properties specify the field's label and control type, for example whether the field is a check box or a text field. You can view and edit the properties for a field on the **Edit Field Properties** dialog box. To access the **Edit Field Properties** dialog box, click on the link of the field that you want to edit in **Issues > Configuration > Custom Issue Tabs** or in **Issues > Configuration > Custom Field Pool**.

You can change the following properties:

| Property | Description |
|-----------------------------------|--|
| Label | The text of the field label, up to 20 characters. Add a colon if you want to separate the label from its associated contents. Special characters are not allowed. |
| Control Type | Active only when user-defined fields are created. The type of control can be Checkbox , Date , Textfield , or Popup List . |
| Case mapping | Active only when the control type is Textfield . Determines the case of the characters that a user enters into a text field. Available values are Upper , Lower , and None . |
| List name | Active only for user-defined popup lists. Select the name of the custom list that defines the values for this popup list. |
| New Issue or Action - Mode | Describes the role of the field when a user reports a new issue. |

| Value | Description |
|------------------|---------------------------------------|
| Required | Must have a value. |
| Optional | Might have a value or might be blank. |
| Read-only | Cannot be edited. |

| Property | Description |
|--|--|
| New Issue or Action - Group permissions | Active when mode is Required or Optional . Permissions defines which groups can enter a value in this field. <ul style="list-style-type: none"> • All gives permission to all groups. • Selected restricts permission to the groups you choose from the list box. When the field is required or optional and permission is restricted to selected groups, the field is read-only for all remaining groups. |

Existing Issue or Action - Mode Describes the role of the field when a user edits an existing issue.

| Value | Description |
|------------------|---------------------------------------|
| Required | Must have a value. |
| Optional | Might have a value or might be blank. |
| Read-only | Cannot be edited. |

 **Tip:** A check box cannot be a required field. If you need a field to be required, consider defining it as a popup list with two opposite values, such as *Yes* and *No*.

| | |
|---|--|
| Existing Issue or Action - Group permissions | Active when mode is Required or Optional . Permissions defines which groups can enter a value in this field. <ul style="list-style-type: none"> • All gives permission to all groups. • Selected restricts permission to the groups you choose from the list box. When the field is required or optional and permission is restricted to selected groups, the field is read-only for all remaining groups. |
|---|--|

Remember value from last issue entered Select this check box to have the **Issue Details** page display the last chosen value. For example, assume that this check box is selected for the **Product** field. When a user reporting an issue first picks a value from the **Product** list box, for example *QA Partner*, that value is preserved, so that the next time the user reports an issue, the **Product** field displays *QA Partner* by default.

Custom Tabs and Fields

Issue Manager permits you to define up to ten user-defined fields per **Custom Tab**.

Custom Tabs

Custom Tab 1 on the **Issue Details** page contains three fields by default. You can add fields to this tab, edit the default fields, or delete the fields and start from scratch.

The following table describes the default fields on **Custom Tab 1**:

| Field | Specifies | Mode | Control Type/Value |
|----------------------|--|----------|----------------------------------|
| Automated? | Whether the issue is automated in regression tests or manual. | Optional | Check box. Default is unchecked. |
| Add Rel Note? | Whether an item on this issue needs to be added to the release notes of the product. | Optional | Check box. Default is unchecked. |

| Field | Specifies | Mode | Control Type/Value |
|-------------|--|----------|----------------------------------|
| Update Doc? | Whether the documentation that is related to this issue needs to be updated. | Optional | Check box. Default is unchecked. |

Custom tabs 2 to 4 are empty.

Custom Fields

Custom fields can appear in the following two places:

- On the custom tabs of the **Issue Details** page.
- On the action dialog boxes, which open when a user takes an action on an issue.

To create or edit custom fields:

1. In the menu, click **Issues > Configuration**.
2. Click **Custom Issue Tabs**. The **Custom Issue Tabs** page opens.
3. Click the custom field that you want to edit. Proceed as described in [GUI Customization Properties](#).

Customizing Workflow

The topics in this section explain the key concepts relating to workflows in Issue Manager. It also presents the three default workflows and describes how they can be edited.

Action-Driven Workflow

Issue Manager manages an issue through its entire life cycle through an action-driven workflow. Action-driven workflow means that an issue is driven from one state (condition) to another by user actions until the issue reaches a terminal, or ending state.



Note: A workflow must have at least one terminal state, which is the last state in the workflow.

Example

In the default workflow, a new bug reported by customer support is considered to be in an *Unreviewed* state, a condition that means that no one has confirmed that the "issue" is truly a bug. Assessing the situation, a QA engineer confirms that the issue is a bug and is ready to send it on to a developer to be fixed. In this example, think of the initial state as *Unreviewed*, the action to be taken as *Confirm*, and the next state as *Dev-Ready*.

If, on the other hand, an identical issue has already been entered, then the action to be taken upon this unreviewed issue is *Mark as Duplicate* and the next state will probably be *Closed*. Therefore a different action upon this unreviewed issue sends the issue to a different state, in this case, *Closed*.

Occasionally, issues retain their current state even after an action has been taken on them. This is true, for example, for *Add Comment*, which allows a user to add a comment to the description of an existing issue.

System-supplied actions

Issue Manager has two predefined actions for each state in each workflow:

- Reassign** Allows users to route an issue to another inbox.
- Edit** Allows users to modify fields on the **Issue Details** page.

These actions do not change an issue's current state, because they do not move the issue through the workflow.

Action Information

A workflow defines a valid set of actions that can be performed on a state. These actions can be viewed on the **Workflows** page. The available actions vary based on the *Issue Type* and *Current State* selected from the lists.

Actions are available on the **Issue Details** page in the form of buttons. The available buttons (actions) vary based on an issue's type and current state.

State Information

State information appears throughout Issue Manager. For example, each group and user account is assigned three initial issue states, one for each issue type: *bug*, *enhancement*, and *documentation* issue.

Initial issue state affects issue life cycle

The initial state of an issue depends on how knowledgeable the user reporting the issue is with respect to this type of the issue. For example, when a member of the Technical Support group reports a documentation issue, it is assumed to be correctly assessed and ready for fixing, and is assigned an initial state of *Open-Doc*. However, when the same person submits a software bug, it is not necessarily assumed to be accurate, so its first state in the workflow is *Unreviewed*. Different groups can have different initial states for the same type of issue. For more information on initial issue states, see "Initial Issue State".

When a user saves an issue, Issue Manager automatically assigns the issue an initial state based on the initial issue state assigned to the user. When a technical writer logs a documentation issue the value of the *State* field on the Issue Details page is *Open-Doc*. The *State* field is an *automatic field*, meaning that Issue Manager, not the user, fills it in based on the workflow and other information.

State Owner

Each non-terminal state in a workflow has exactly one owner. The owner is the role in an organization that is responsible for acting on an issue in a given state. Consider an unreviewed bug: the user who confirms or denies an unreviewed software issue is most likely to be someone who performs the QA role. Therefore, the state owner of an unreviewed bug is the QA role.

A terminal state in a workflow does not have an owner because an issue in this state does not need someone to be responsible for taking an action on it.

In Issue Manager you choose one of four possible owners for a non-terminal state:

- QA
- Development
- Documentation
- Enhancements

 **Note:** A state owner is not a specific QA engineer or a specific inbox; nor is it related to a specific product, component, release, or platform. A state owner is a general designation of functional responsibility with respect to a state.

The owner is an important property of a state because the state owner and the routing rules together determine the specific inbox that will receive the issue (routing rules are described in *Setting Up Routing Rules*). Here is an example of how Issue Manager uses routing rules, states, and state owners to move an issue to a specific inbox.

Say that you decide that the owner of the *Unreviewed* state of all software bugs, regardless of specific product, component, and so on, should be the QA role. Individuals who fulfill this role will confirm that a reported issue is actually a bug. So, on the *State Properties* dialog for the *Unreviewed* state, you select the *QA Owns This State* radio button. To access this *State Properties* dialog, go to **Issues > Configuration > Workflows**, and then click **Edit State**.

Now consider routing rules for specific products. When you set up routing rules (**Issues > Configuration > Routing Rules**), you specify four specific inboxes for each combination of product, component, release,

and platform. Each of the four state owner radio buttons corresponds to one of the four inboxes on the **Routing Rules** page: *QA*, *Development*, *Enhancement*, and *Documentation*.

Example

A routing rule states that when a bug pertains to the *Email* component in any release of *Product C* on any platform, then route the bug to one of these inboxes: *Mike - QA*, *Sonja - Dev*, *Dan - Dev (Product C)*, or *Judy - Doc*.

One of the four inboxes is selected based on two factors: the current state of the issue and the owner of that state. Assuming that the current state of the bug is *Unreviewed* and that you specified that the QA role owns unreviewed issues, then the issue will be automatically routed to Mike's inbox, *Mike - QA*.

When Mike acts on the issue, he will, in effect, move it along its life cycle to another state, which has another owner. Issue Manager will again determine the appropriate inbox based on the issue's current state, the owner of the state, and the applicable routing rule for the specific product, component, release, and platform.

Reason Codes

Issue Manager reason codes are optional, customizable keywords that describe why an issue has changed its state when a given action is taken.

A number of actions can cause an issue to move to the same state. For example, an issue can be closed for a variety of reasons: it's not reproducible, it's a duplicate, it's not a bug. Without the extra information supplied by the reason code, users will have an incomplete picture of an issue's life cycle.

Reason codes can also help you minimize the number of states in your workflow. For example, instead of defining several terminal states- *Not a Bug*, *Not Reproducible*, *Duplicate*, it is sufficient to have one terminal state called *Closed* with a variety of reason codes that indicate why an action closed an issue with, for example, *Closed/Not a Bug*.

The user sees reason codes on the **Issue Details** page and the action dialogs. For example, Judy, a technical writer, receives a documentation issue in her inbox. Reading the description, she recalls that the issue has already been reported. She marks the issue as a duplicate. When the **Mark as Duplicate** dialog box opens, she can see that the issue has moved from *Open-Doc* to *Closed/Duplicate*. *Closed* is the new state and *Duplicate* is the reason code.

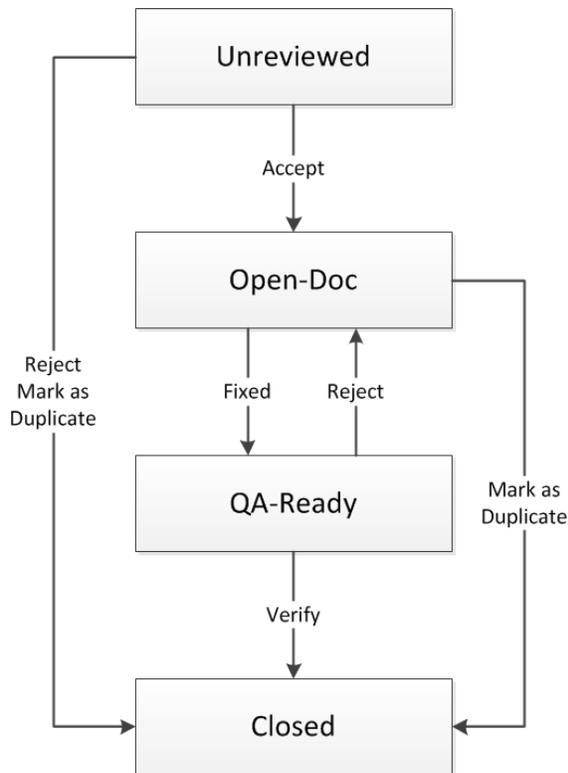
Assigning, clearing, or retaining a reason code

When certain actions are taken, Issue Manager assigns a reason code to the action and passes it on to the new state. Subsequent actions might clear the reason code or simply retain the reason code. In general, once you set a reason code, it travels with the issue until it reaches the terminal state in the workflow.

Consider the case of a developer who fixes a bug and then takes the *Fixed* action to claim that the bug has been fixed. This action moves the bug from *Dev-Ready* to *QA-Ready*, and sets the reason code to *Fixed*. The QA engineer who is verifying the developer's claim accepts the bug fix by taking the *Verified* action. This action retains the *Fixed* reason code and moves the bug along to the *Closed* state.

However, you might want to clear the reason code, for example, when an issue returns to an earlier state in the workflow, instead of progressing toward a terminal state. Say that a QA engineer disputes a developer's claim and takes the *Reject* action, which sends the issue back to the *Dev-Ready* state. The *Fixed* reason code no longer makes sense, so you might want to clear the reason code for the *Reject* action.

Whether reason codes are assigned, cleared, or retained is determined by the setting in the **New Action for State** dialog, which is described in Reason code. Click **Issues > Configuration > Workflows** and click **Add Action** to view the **New Action for State** dialog box



The *Edit*, *Reassign*, and *Add Comment* actions have been omitted from this diagram.

- *Edit* and *Reassign* are predefined for each state and cannot be modified. These actions do not change an issue's state.
- *Add Comment*, which is defined for all states in this workflow except *QA-Ready*, does not change an issue's state.

Default reason codes in the documentation workflow

To see all the reason codes supplied in the default documentation workflow, click **Issues > Configuration > Workflows**. Select `DOC-ISSUE` as the issue type and view its valid actions and reason codes.

Example

Assume that a documentation issue enters in the *Unreviewed* state. At that point there are three possible actions that the reviewer can take. He can accept the issue, reject the issue, or mark it as a duplicate of another issue. If the issue is rejected or found to be a duplicate, then it will be closed, with a reason code of *Rejected* or *Duplicate*. If the reviewer considers the issue a documentation issue that needs to be fixed, then the issue will be moved to the *Open-Doc* state and to a documentation inbox. The documentation specialist can take one of two actions at this point: either take the *Fixed* action or the *Mark as Duplicate* action.

- The *Fixed* action sends the issue to the *QA-Ready* state with a reason code of *Fixed*.
- The *Mark as Duplicate* action sends the issue to a *Closed* state with a reason code of *Duplicate*.

If the fix is later verified, then the issue will be moved to the *Closed* state with a reason code of *Resolved*. If, on the other hand, the fix is rejected, then the issue will be returned to a documentation inbox with a reason code of *Rejected*.

Developing Your Own Workflows

If you want to develop your own workflow, the easiest and most reliable approach is to make simple modifications to the default workflows (for example, edit a state name). It is strongly recommended that you

modify the default workflows rather than create a new workflow from scratch, especially if you have little or no experience with these concepts.

Some of the things that you can customize include:

- The number and name of the states that an issue passes through.
- Which actions are valid on a particular state. These actions appear on the Workflows window.
- Which states result from each action on the current state; that is:

```
Current State ---- Action 1 ----> New State 1
Current State ---- Action 2 ----> New State 2
```

- Which groups have permission to perform each action.
- The appearance of the action dialogs that are displayed when users take action on issues.

Prerequisite

Familiarize yourself with the concepts of state diagrams and actions, also called state transitions.

Drawing a State Diagram

Draw a state diagram on a whiteboard or paper. Make sure to draw each state in a box with plenty of space between each state.

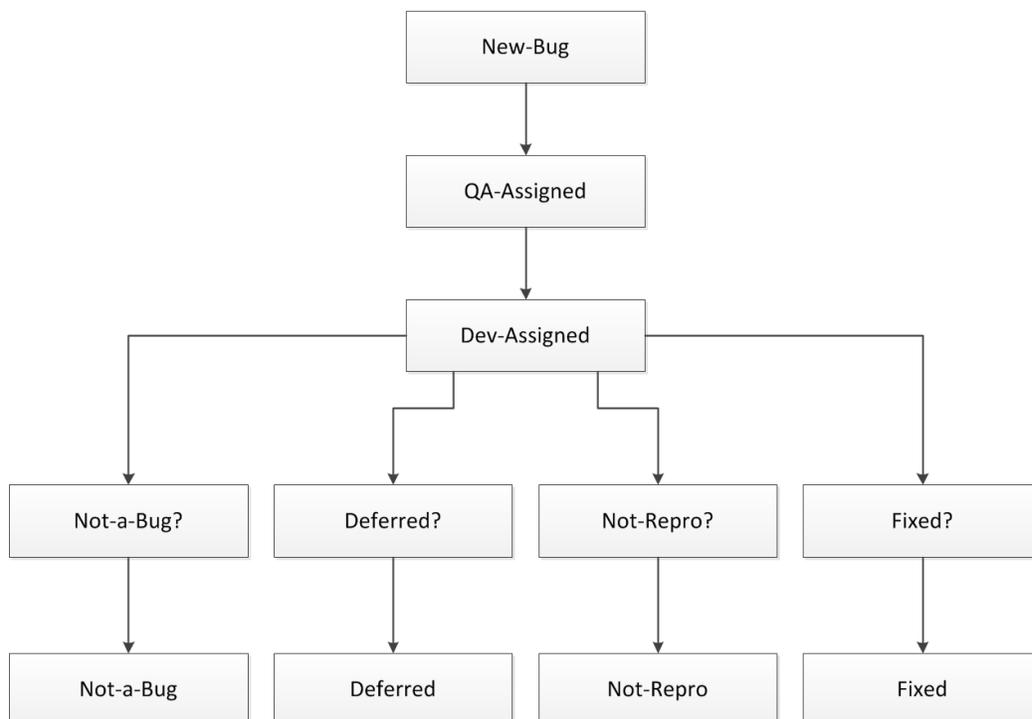
Draw each legal action between states as a line with a single arrow head. Each direction is a distinct action.

Example

The following example is a small sample of the states you might have in your own workflow. When a new bug is entered, someone in your organization dispatches the bug to the appropriate QA engineer for review. If the QA engineer agrees that the reported behavior is a new bug, then it is sent to Development to be fixed. Development can take a number of possible actions on the bug.

- The behavior described in the bug report is designed behavior, so the issue is not a bug.
- The bug should be deferred until the next release.
- The bug cannot be reproduced.
- The bug is fixed.

Of course, these are only the developer's claims and have yet been verified. Your workflow model might reflect these claims with the following states: *Fixed?*, *Deferred?*, *Not-Repro?*, and *Not-a-Bug?* The question marks imply that these claims have not yet been verified by a QA engineer.

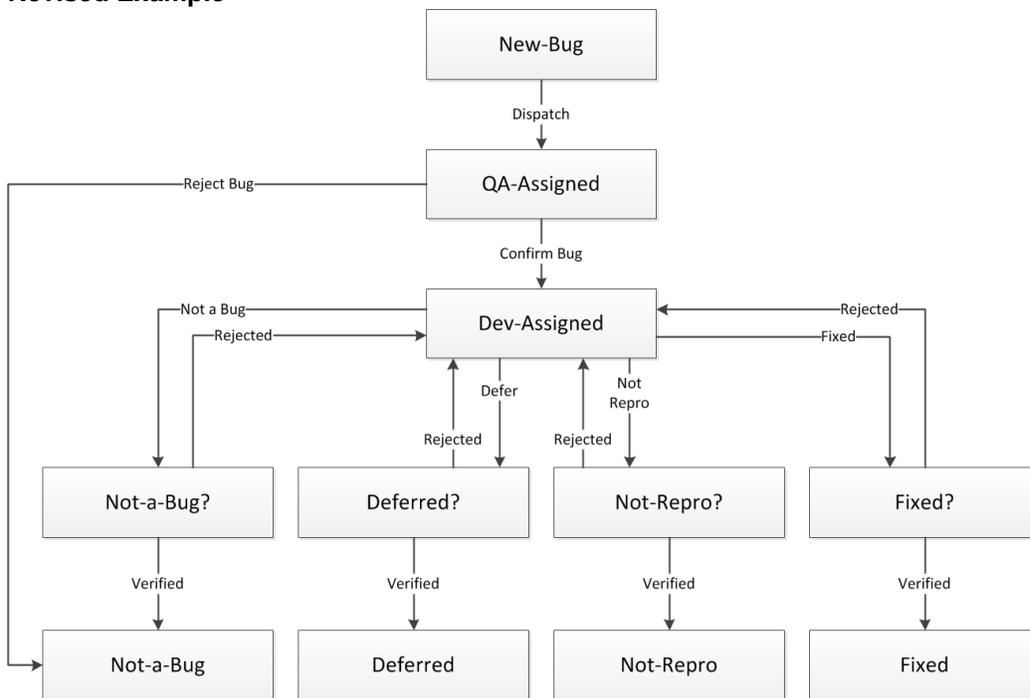


Name Actions with Verbs

It is recommended that you assign a verb to each action. Think about times when a bug might return to an earlier state in the workflow. Model these negative actions. For example, a QA engineer might reject a developer's claim, which returns an issue to Development.

You may also uncover missing positive actions while considering each way that an issue can travel. For example, you realize that QA engineers also reject some issue reports, in effect closing the issues. In the example diagram below, the Reject Bug action by a QA engineer leads to the Not-a-Bug state. Ensure that all actions eventually lead to a terminal state.

Revised Example



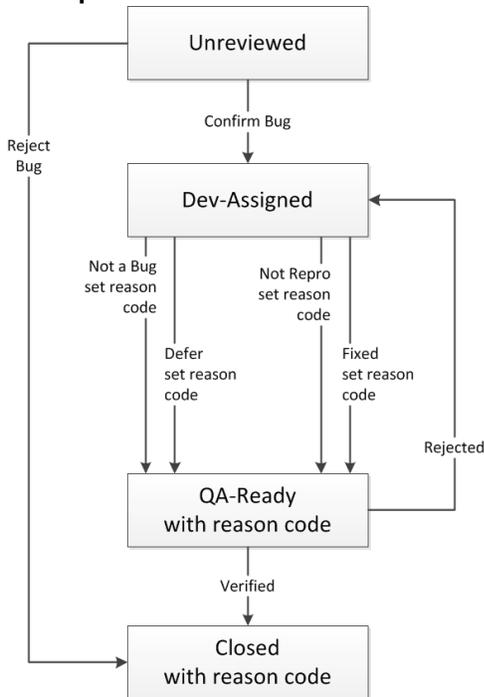
Optimizing the Workflow

As you can see, the preceding diagram is too complicated to be useful. To simplify the workflow you can eliminate the Dispatch action because the routing rules in Issue Manager take care of distributing bugs to the correct inboxes. In the preceding diagram you can delete the New-Bug state and the Dispatch action and start the bug in a state where a QA engineer can review it. You might call this the Unreviewed state.

Another flaw in the workflow is the virtually repetitive states. Using Issue Manager's methodology, you can eliminate redundant states by assigning reason codes to actions. For each action that requires a reason code, devise a brief, descriptive keyword. For a full explanation of reason codes, please read "Reason codes". To identify redundancies, look for repeated patterns in the workflow. For example, in the previous diagram, the last row of states (Not-a-Bug, Deferred, Not-Repro, and Fixed) can be collapsed into the single state of Closed with a different reason code for each action (for example, Closed/Not-a-Bug, Closed/Fixed).

The next-to-last row can also be collapsed into a single state that recognizes QA's role in verifying Development's claims. You might call this state QAReady. You can use reason codes here to explain why a bug has changed states (for example, a bug arrives in the QA-Ready state with a reason code of Fixed or Deferred). The four Rejected actions taken by QA can be collapsed into a single Rejected action. Similarly, the four Verified actions taken by QA can be collapsed into a single Verified action.

Example



Add State Owners and Permissions

Issues are routed according to issue state and state owner. A state owner is a role in your organization that is responsible for an issue in a given state. For example, in the default workflow, QA has responsibility for issues in the Unreviewed state.

Now you must assign an owner to each state in your final diagram. Finally, decide which groups should have permission to take each action. In the default workflow, for example, only users in Development are allowed to fix issues (that means, take the Fixed action).

Prepare a Data Entry Sheet

Optionally, prepare a sheet to facilitate data entry into Issue Manager. The workflow information is entered by state, so the first column should be current state and owner. Then add the following column headings:

- Actions allowed on the current state.
- New state resulting from each action.
- Reason code (if applicable).
- Permissions - The group(s) that are allowed to take each action.

Your data entry sheet should resemble the following:

| State/Owner | Actions | New State | Permissions |
|-------------|---------|-----------|-------------|
| | | | |

Workflow State Customization

You can modify the names, owners, and descriptions of the states in the default workflows. You can enter or modify all states at once and then add all actions for each state. Or you can enter one state and all its actions before going on to the next state and its actions.

State Properties

States have the following properties:

State Name Required, unique state name, up to 20 characters long. Special characters are not allowed.

Description Optional description of the state; no practical length limit.

State Owner Choose a state owner by selecting the appropriate option:

- QA Owns This State
- Development Owns This State
- Documentation Owns This State
- Enhancement State
- No One (Terminal State)

Customizing Workflow States

You must have the `workflow customization` security privilege to perform these tasks.

1. Click **Issues > Configuration > Workflows**. The **Workflows** page is displayed.
2. Select the **Issue Type** that represents the workflow you want to customize.
3. Proceed with tasks as outlined below:

Add a state to this workflow

Click the **Add State** button. Proceed to step 4.

Edit a state in this workflow

Select the state you want to edit from the **State** list. Click the **Edit State** button. Proceed to step 4.

Add actions for a state in this workflow

See *Customizing Actions, Reason Codes, and Action Dialogs*. Begin the procedure at step 3.

Delete a state in this workflow

Select the state you want to delete from the **State** list. Click the **Delete** button. Select **Yes** on the confirmation dialog. Issue Manager will only delete states that have never been referenced by an issue.

Modify another workflow

Return to step 2.

4. Specify or modify the properties of the state, using the information described in *State properties* to guide you.

Customizing Actions, Reason Codes, and the Action Dialogs

Read this section only if you intend to modify the actions in the default workflows, which are described in *Understanding the Default Workflows*. Before proceeding, please read *Action Properties*.

1. Click **Issues > Configuration > Workflows**.
2. Select the workflow you want to customize from the **Issue Type** list.
3. Select the state that has the actions you want to modify from the **State** list.
4. The **Workflows** page now illustrates the information you will enter in this procedure:
 - All the actions that apply to this state (as they appear in the **ButtonLabel** column).
 - The state that follows the current state when a given action is taken.
 - The reason code for each valid action (if applicable).



Note: The Edit and Reassign actions, which Issue Manager predefines for all states, cannot be modified and are therefore not shown in the dialog.

5. Proceed with tasks as outlined below:

| | |
|--|--|
| Add an action for the current state | Click Add Action and proceed to step 6. |
| Edit an action for the current state | Select the action you want to edit from the Button Label column. |
| Add a new state to this workflow | Return to step 3 of <i>How to Customize workflow states</i> . |
| Change the sort order of an action in this workflow | Click the Move Up and Move Down buttons (in the Actions column) of the action you want to move up or down within the list. |
| Delete an action from another workflow | Click the Delete button (in the Actions column) of the action you want to delete. Then click Yes on the confirmation dialog. |
| |  Note: Issue Manager will delete only those actions that have never been referenced by an issue. |
| Modify the actions in another workflow | Return to step 2. |
| Modify the states in another workflow | Return to step 2 of <i>How to Customize workflow states</i> . |

- Specify or modify the properties of an existing action on the **New Action for State** dialog box. Use the information in *Action Properties* as a guide.

Action Properties

After you define the states in a workflow, you must define the actions that are associated with each state. For example, for the *Dev-Ready* state in the default bug workflow, a developer can take the *Fixed* action, the *Cannot Fix* action, and so on.

Each action has a set of properties associated with it.

General properties

The properties of an action are varied and cover the following areas:

- **Button label** that appears on the **Issue Details** page.
- Description of the action that appears on the **History** tab.
- New state that results from this action.
- Reason code.
- Appearance and usage of the action dialog.
- Groups permitted to take this action.
- Sort order of this action.

Action of State dialog box

Action properties are set on the **Edit Action of State** dialog box. (**Issues > Configuration > Workflows**).

Current State Information

| | |
|---------------------|---|
| Name | Cannot be edited. Displays the name of the current state for which you are defining or modifying an action (for example, <i>Edit Action for State - Unreviewed</i>). |
| Button label | This is the name that appears on the button that executes this action, located on the Issue Details page, up to 30 characters long. |

| | |
|----------------------------|--|
| History Action Code | Concise statement of the action that was taken, up to 20 characters long. It is recommended that you use uppercase characters (for example, VERIFIED). This code appears in the Actions column on the Issue Details page's History tab. |
| Tooltip | Optional text that will appear as a tool tip when a user passes their cursor over this action button. |
| Description | Optional description of this action. |

New State List

| | |
|--------------------|--|
| Name | Name of the next state in the workflow when this action is taken on an issue. For example, in the default workflow when an <i>Unreviewed</i> issue is confirmed as a bug, Issue Manager moves the issue to the next state, <i>Dev-Ready</i> . |
| Reason Code | Optional keyword that describes why an issue has changed from the current state to the new state when this action is taken. Select the appropriate option: <ul style="list-style-type: none"> No Change No Change indicates that the keyword is retained when the issue moves to the new state. It will appear in the Reason Code field of the Issue Details page. For example, the following workflow shows that when a developer claims to have fixed a bug, the bug moves to the QA-Ready state with a reason code of Fixed. If a QA engineer verifies the claim, then the bug moves to the Closed state while retaining the Fixed reason code. Any user browsing issues can easily see the reason the bug has been closed by looking at the reason code. Clear Indicates that the reason code for the current state will be removed when the issue moves to the new state. Clear is a reasonable choice when an issue returns to a previous state in the workflow (for example, when a developer claims to have fixed a bug, the bug moves to the <i>QA-Ready</i> state with a reason code of Fixed). If, however, the “fix” is rejected, the issue is sent back to Development (<i>Dev-Ready</i>) and the <i>Fixed</i> reason code is removed, since this claim is disputed. When you choose Clear, the workflow displays CLEAR in the Reason Code field; however, the user sees an empty Reason Code field on the Issue Details page. Set to Indicates that you can associate a reason code with this action. Enter a keyword of up to 20 characters. All capital letters is recommended. Use Set To to specify a reason code when an action moves an issue to a new state that requires a reason code. In general, you should assign reason codes to all actions that developers take. For example, in the default workflow the <i>Fixed</i> action on the <i>Dev-Ready</i> state sends the issue to the <i>QA-Ready</i> state with a reason code of <i>Fixed</i>. What this means in terms of human behavior is that when a developer claims that a bug has been fixed, he hands it off to a QA engineer, who can now easily scan the Issue Details page to see why the bug is in his or her inbox (the bug could also be there because the developer can't fix it or the software is working as designed). |

If you do not use reason codes

Reason codes are optional keywords that can help to minimize the number of states in your workflow. For example, rather than defining several closing states (*Not a Bug*, *Not Reproducible*, *Not Repro*, and *Duplicate*) one terminal state is sufficient where reason codes help explain why the issue is in the *Closed* state. If you decide not to use reason codes, you may need to have several terminal states. A state can be made a terminal state by selecting the radio button called **No one** (Terminal State), which appears as a choice in the **State Owner** field of the **State Properties** dialog.

Standard Action Fields Tab

The **Standard Action Fields** tab on the **New Action for State** dialog defines the properties relating to the use and appearance of the action dialog, specifically:

- The **Action Notes** field on action dialogs.
- **Release Information** on action dialogs.
- **Related Issue Number** of a related issue on action dialogs.

Action Notes

- Not Used** The **Action** dialog box will not display the **Action Notes** field when the user takes this action.
- Optional** The **Action** dialog box will display the **Action Notes** field. Users have the option of entering extra information about the action they are taking.
- Required** The **Action** dialog box displays a required **Action Notes** field. Users must enter extra information about the action in this field.

Release Information

This group allows you to have a list appear on the action dialog from which users can select the release in which this action is taken. The release selected by the user also appears in one of the automatic fields on the **Issue Details** page. This information is particularly useful for certain kinds of actions (for example, those related to confirming, fixing, and verifying issues). It is not useful for actions that do not change the current state. The **Release Information** field has one of four values:

Not Used The action dialog will not display the list. The action dialog for the **Add Comment** action does not have a list box for release information because release information is not necessary for the **Add Comment** action.

If you choose **Not Used**, for example if the user doesn't supply information in an optional list box on the action dialog, or if an action hasn't been taken which uses the list, then the **Issue Details** page displays an empty **Action Release** field.

Optional or Required The action dialog will display an **optional** or **required** list from which the user selects a release. The last automatic field on the **Issue Details** page displays the release in which the action was taken. You need to specify the list **Label** to make it appropriate for the action. The label can be up to 20 characters long, including a trailing colon. The default field label is **Confirmed In:** for confirming actions; **Fixed In:** for fixing actions; and **Verified In:** for verifying actions. The label appears on both the action dialog and on the **Issue Details** page. The list's values derive from the list of releases specified in the Administration module.

Cleared The previous value in the last automatic field of the **Issue Details** page will be removed and the field label reverted to the default, **Action Release**. The **Label** field in the **Release Information** group is disabled. **Cleared** is a good choice when an action causes an issue to return to a previous state in the workflow. Examples from the default workflow are the **Reject** and **Reopen Bug** actions. For example, a developer takes a *Fixed* action and fills in a specific release number in the **Fixed In** field. This action sends the issue to the *QA-Ready* state. The QA engineer responsible for the issue rejects the developer's claim that the bug has been fixed. The **Reject** action moves the issue from its current state, *QA-Ready*, back to a previous state in the workflow, *Dev-Ready*.

When an action's reason code is cleared, consider choosing **Cleared** as the value for the **Release Information** field.

The release selected by the user on the action dialog will appear on the **Issue Details** page. For example, say that a user takes the Fixed action, selects 4.1:prod as the release from the Fixed In list box, and clicks OK. When the Issue Details page reappears, the Fixed In field displays the selected release.

Related Issue Number Allows you to place a text field on the action dialog in which users can enter the number of a duplicate or related issue. This information is useful when users take the *Mark as Duplicate* action in the bug workflow or the *Already Done* action in the enhancement workflow.

Not Used Text field will not appear on the action dialog.

Optional or Required An optional or required text field appears on the action dialog. You need to specify the text field label. The label can be up to 20 characters long, including a trailing colon. The default field label in the bug workflow is **Duplicate of #**. The default field label in the enhancement workflow is **See Also**.

The issue number specified by the user on the action dialog will appear in the **Notes** field of the **Issue Details** page's **History** tab. For example, say a user takes the *Mark as Duplicate* action on Issue #8, specifies Issue #6 as the duplicate, and clicks **OK**. When the **Issue Details** page reappears, the **Notes** field will display the phrase *Related to issue number n, where n is the value of the Duplicate of # field*. If the **Duplicate of #** field is optional and the user does not fill it in, then the **Notes** field will display the contents of the **Action Notes** field.

User-Defined Action Fields Tab

The **User-defined Action Fields** tab of the **New Action for State** dialog allows you to add fields to the action dialog of a given action.

Eight field positions These fields appear on the action dialog in addition to the standard action fields. For example, the **Hours to Fix** field might be selected from the list to appear on the action dialog for the *Fixed* action in the *Dev-Ready* state.

Sources The fields that appear in the list boxes on the **User-defined Action Fields** tab are there for one of two reasons:

- Custom fields were defined through **Issues > Configuration > Custom Issue Tabs** to define custom fields which are also visible on the issue details **Custom** tab.
- Custom fields were defined through **Issues > Configuration > Custom Field Pool**, which are only relevant for actions.

Defining Fields on an action dialog

1. Click **Issues > Configuration > Custom Issue Tabs** to define custom fields which are also visible on the issue details **Custom** tab. You can also use custom fields that were created through **Issues > Configuration > Custom Field Pool**, which are only relevant for actions.
2. Configure the field properties as explained in [GUI Customization Properties](#).
3. The custom fields you create can then be selected from the list boxes on the **User-defined Actions Field** tab.

- For each defined field, you can select an option from the list to specify whether a field is required and whether the field should be cleared before a tester performs the action. By default, fields stay as they were configured during definition and are not cleared.



Note: Check boxes cannot be set to be required and cannot be cleared.

| Option | Description |
|------------------|--|
| --- | The field stays as configured during definition, required or not required, and is not cleared. |
| Clear | The field stays as configured during definition, required or not required, and is cleared. |
| Required | The field is set to be required and is not cleared. This option is only available if the field was not already set to be required when it was defined. |
| Clear & Required | The field is set to be required and is cleared. |

Permissions Tab

On the **Permissions** tab you can set security for individual actions, thus restricting actions to selected groups of users.

By default, **Apply Security** has a value of **No** (that means, all groups have permission to take all actions). This setting appears on the **Workflows** page as (all groups). You set security by selecting the **Yes** option and then selecting one or more groups to have permission to take the action.

Consider this example of defining security for an action: When a developer opens an **Issue Details** page for a bug in the *Dev-Ready* state, the *Cannot Fix* action is available. However when a user who is not a developer opens the **Issue Details** page for the same bug, the action is hidden.

Intelligent Assistant

To process test results, the Intelligent Assistant uses a default set of four state/reason code pairs that correspond to the four general states defined by the default workflow for bugs.

To view these default states and reason codes:

- Click **Configuration > Intelligent Assistant**.
- Customize mapping between general bug states and specific Issue Manager codes.

If the default workflow for issues has been customized, the states and reason codes used for results processing must be customized too. For example, if the administrator changes the default workflow so that issues claimed to be fixed are assigned a state of *Unreviewed*, he must also change the state entry for *Bugs Claimed to Be Fixed But Not Tested Yet* from *QA-Ready* to *Unreviewed*.

In addition to the issue states and reason codes specified on the Intelligent Assistant page, Intelligent Assistant also uses the terminal states defined for the issue workflow. Terminal states are defined on the Workflows page. The default terminal state is *Closed*.

Issue Archival

After using Issue Manager for a while, you may find that you have issues in the database that are no longer relevant to your efforts (for example, issues that are closed or pertain to unsupported products). The more issues in the database, the longer actions (such as queries) take to execute. To help you improve performance, you can archive issues. You can then run actions against the smaller number of active (not archived) issues.

When you archive issues, Issue Manager moves the issues from the `DEFECT` and related tables to the `ARCHIVE` and related tables. All issue information is preserved. (By comparison, active issues remain in the `DEFECT` table, and users can take the usual actions on them.)

Users may view archived issues on the **Issue Details** page but they cannot take actions on them because they are read-only. Archived issues do not appear in users' inboxes. However, users may still run queries, reports, and graphs against archived issues. Archiving "hides" issues in a database, but it does not delete them.



Note: It is strongly recommended that you not delete issues manually, as this may destroy the integrity of your data.

You may however, want to make your production database smaller by creating a new custom database that imports only the active issues and related information from the original database.

If you archive an issue that is being displayed (but not edited) on a user's desktop, the user won't know about the change until they log out and log back in again, exits and restarts, or tries to take an action on the issue.



Note: Issue Manager warns users that they cannot take action on archived issues.

Archiving Issues

To archive issues:

1. In the menu, click **Issues > Configuration > Archive**.
2. Select the product that relates to the issues you wish to archive.
3. The **Release** field is automatically populated with the releases related to the selected product. Select the specific release that relates to the issues you wish to archive.
You can archive issues relating to multiple releases by selecting multiple releases. Use **Ctrl+Click** to select individual releases, or **Shift+Click** to select a range of releases.
4. Click the **Archive** button.
5. Click **Yes** on the confirmation dialog to initiate the archiving process.

Setting Up Email Notification Rules

With email notification, you can instruct Issue Manager to notify you or others in your organization automatically when events of interest occur, such as when an issue changes status.

Email notification rules

Users can have email sent to them whenever the following events occur:

- An issue is reassigned to another inbox.
- An issue moves from one workflow state to another (for example, from `Unreviewed` to `Dev-Ready`).
- Any change is made to an issue that generates a new entry in the issue's history (for example, the component against which the issue is reported has changed).
- Any other special event that you define (for example, Technical Support wants to receive email when a particular bug is fixed).

The events listed above are referred to as rules. The first three rules are predefined and cannot be modified. You may define as many additional rules for special events as you want.

Notifications

An email notification instructs Issue Manager to send email whenever a certain event occurs to an issue. A notification consists of an email notification rule that specifies an event, the sender and recipients of the email, the message body, and an optional subject line, introduction, and trailer.

Users must apply a notifications to each issue they are interested in. Only users with the **Email notification per issue** privilege can add notifications to individual issues.

To view an issue's notifications, click the **Notifications** icon on the issue's **Issue Details** page.

Rules can also be used in notifications that apply to all issues in the project; these are project-wide notifications.

Only users with the **Manage email notifications** security privilege can add project-wide notifications.

Defining Email Notification Rules

Prerequisites

To define your own rules, you must:

- Have the **Manage email notifications** privilege.
- Be familiar with SQL, specifically how to compose a WHERE clause.
- Be familiar with the database schema, in particular, the columns of the issue-related view and table, `IM_V_Defects` and `IM_DefectHistory`.
- Be familiar with your workflow and the values in the workflow.

Tips for writing SQL WHERE clauses

Here are some important tips that will help you write syntactically and semantically correct WHERE clauses.

Reference both `IM_V_Defects` and `IM_DefectHistory`

Your SQL WHERE clauses will probably require references to both `IM_V_Defects` and `IM_DefectHistory`. The `IM_V_Defects` view stores information that is current for an issue, whereas the `IM_DefectHistory` table keeps a record of all actions that have been taken on an issue, along with the effects of those changes on some issue fields.

For example, `IM_DefectHistory` stores an issue's inbox assignment before an action is taken as well as the inbox assignment after the action. These columns are **AssignedIN** for the inbox assignment prior to the action and **AssignedOUT** for the new inbox.

All actions taken on issues are recorded in the **ActionCode** field of the `IM_DefectHistory` table. These actions appear as action codes in the **Action** column of the **History** tab. You may have noticed such codes as FIXED and VERIFIED in the sample database.

To see the action codes for most of the actions in your database, look at the **Edit Action of State** dialog box of each state (exceptions are ENTERED, REASSIGNED, and MODIFIED, which are hard-coded and cannot be viewed). To access the **Edit Action of State** dialog box for a state, go to **Issues > Configuration > Workflows**, click the name of a state in the **Button Label** column. Look at the value in the **History Action Code** field.

Use aliases

The views and tables have been aliased. You must use the alias `D` to refer to the `IM_V_Defects` view and the alias `DH` to refer to the `IM_DefectHistory` table.

Identifying custom fields

Custom fields are identified in the `IM_V_Defects` view as `Custom1`, `Custom2`, and so on, depending on their position on the tabs. Each custom tab has up to 10 fields, 1-10, 11-20, and so on. On **Custom Tab 1**, the first five fields appear in descending order in the left column; field 6 through field 10 appear in descending order in the right column.

To find out the schema name of a particular custom field, go to **Issues > Configuration > Custom Issue Tabs**. For example, in the dialog for the sample database, the **Add Rel Note?** check box is the fourth field in the left column of **Custom Tab 1** and so is **Custom4** in the issue table.



Note: If you change the position of a custom field, then you will need to update any email notification rules that refer to the field.

Changes to custom fields are not tracked

Issue Manager does not track changes to the custom fields in the `IM_DefectHistory` table. You can check the current value of a custom field, but you cannot refer to a previous value. For example, the `WHERE` clause can test whether or not the **Add Rel Note?** check box is selected, but it cannot capture a change in the value of the check box, from unchecked to checked.

Accessing check box values

The value of an unselected check box is '.' (a period). The value of a selected check box is 'X' (capital X).

For example, to retrieve all issues where **Add Rel Note?** is checked, you would specify as part of the `WHERE` clause:

```
D.Custom4 = 'X'
```

Examples of rules

Here are four situations in which you might want to create a new rule. The first two situations suggest rules used in notifications on individual issues. The last two situations suggest rules used in project-wide notifications. Following each example is the `WHERE` clause written against the sample database.

Example 1 Technical support and other groups want to know when a particular bug fix has been verified as fixed.

The `WHERE` clause looks like this:

```
DH.ActionCode = 'VERIFIED'
```

The value of **ActionCode** in the `IM_DefectHistory` table is updated whenever a user takes an action; therefore, Technical Support will receive mail only once, when the `Verify` action causes the action code `VERIFIED` to be entered in the `IM_DefectHistory` table and the History tab.

If the clause were written instead by referring to the **Reason Code** (known as `Disposition` in the database):

```
D.Disposition = 'FIXED'
```

then Technical Support would almost certainly receive mail more than once, because the Reason Code remains `FIXED` in the database. A user who subsequently adds a comment or saves the issue would trigger email because the rule would still match.

Example 2 An average user, one who does not act on bugs, would like to know what has happened to a bug that he entered. In particular he wants to receive email when the bug has received a developer's attention. In workflow terms, this means the issue has just left the `Dev-Ready` state, and consequently, the `WHERE` clause must test for state change after the action.

The `WHERE` clause is:

```
DH.StatusIN = 'Dev-Ready'  
AND DH.StatusOUT <> 'Dev-Ready'
```

The state was `Dev-Ready` before the action, but after the developer's action, the bug moved to another state.

Example 3 The documentation department has all doc-issues sent to a group inbox, **Doc (Product A)**, rather than a user's inbox, **Judy -- Doc**. The documentation manager wants to receive email when an issue enters the group inbox.

This WHERE clause is:

```
DH.AssignedIN <> 'Doc (Product A)'  
AND DH.AssignedOUT = 'Doc (Product A)'
```

It specifies that email should be sent when the inbox before the action was not **Doc (Product A)** but after the action is **Doc (Product A)**. The email will be sent only once, when the doc-issue is routed to **Doc (Product A)**.

Note that without the first part of the WHERE clause, the documentation manager would receive email when any action was taken on the doc-issue while it was assigned to Doc (Product A). Although the rule is likely to generate a large volume of mail, since the notification needs to be applied project-wide, the two lines together of the WHERE clause restrict the rule to single event: when the inbox becomes **Doc (Product A)**.

The rule does not need to specify the issue type to be DOC-ISSUE, although it would not be incorrect, since **Doc (Product A)** only holds documentation-related issues.

Example 4 The release manager wants to receive email during the next month prior to a major release regarding the most severe bugs that cannot be fixed. The WHERE clause must test for severity, Product B, and the Cannot-Fix action code assigned when a developer takes the Cannot Fix action.

The WHERE clause is:

```
D.Severity = '1: Fatal/Data Loss'  
AND DH.ActionCode = 'Cannot-Fix'  
AND D.ProductCode = 'Product B'
```

These restrictions are necessary because if the WHERE clause merely tests for severity, then the rule will match anytime the issue is changed and saved, because the severity will not change until a user explicitly changes it.

The release manager might receive a great deal of mail, especially if this rule is applied in a project-wide notification. However, he can easily delete the notification after the release cycle is over.

Tips for writing WHERE clauses

Work from the general to the specific. First, consider the general business situations that might require email notification rules. You might ask all Issue Manager users when they would like to receive email about issues. Find out precisely what information each user wants to glean from the email. For example, a user might tell you that he wants to know when a bug is fixed. Upon further discussion you might find that what he really wants to know is when the fix is verified by a QA engineer. This subtle change might require a different WHERE clause.

Then, when you're satisfied that you understand what users want, translate the situation in terms of your organization's workflow.

Finally, write the SQL WHERE clause. Try testing the WHERE clause through an advanced query to make sure you are specifying the conditions exactly as intended.

Generally speaking, rules intended for notifications on individual issues should be written simply and generically, whereas rules for project-wide notifications should be as precise and as restrictive as possible to avoid excessive email.

Ask yourself how often users want to receive notifications; only once or each time a change is generated. If a user wants mail only once, then make the rules more restrictive.

Managing Email Notification Rules

You need the **Manage email notifications** security privilege to add, edit, or delete email notification rules.

To add, edit, or delete an email notification rule:

1. Click **Issues > Configuration > Notification Rules**.

2. Based on the task you wish to perform, proceed as outlined below:

| | |
|---------------------------------------|---|
| Add a Rule | Click the Add Rule button. The New Notification Rule dialog is displayed. Proceed with step 3. |
| Edit a Predefined Rule | Predefined rules cannot be edited. |
| Delete a Predefined Rule | Predefined rules cannot be deleted. |
| Edit a Rule you have created | Click the Edit icon (in the Actions column) of the rule you wish to edit. Proceed as explained in step 3. |
| Delete a Rule you have created | Click the Delete icon (in the Actions column) of the rule you wish to delete. Click OK on the confirmation dialog to execute the deletion. |

3. Specify or modify the following properties on the **New Notification Rule** dialog box:

| | |
|---------------------|---|
| Rule Name | Required unique name, up to 30 characters. Make the name as descriptive as possible, as users assign rules by name. |
| Description | Optional description of the triggering event, up to 250 characters. |
| WHERE clause | Required SQL WHERE clause that specifies the triggering event, up to 250 characters. Do not type the WHERE keyword. |

4. Click **OK** to save your settings, or click **Cancel** to exit without saving.

Once you have defined rules, make sure that you:

| | |
|---|---|
| Specify email addresses for all users | To make use of email notification, a user must have a valid email address specified in his or her user account. |
| Have your database administrator enable email notification | Once the feature is enabled, the Notifications icon appears on the Issue Details page. |
| Educate your users | Inform your users as to the rules that are available, and what each rule accomplishes. Users can view the rules by selecting Issues > Configuration > Notification Rules . |

Deleting an Inbox

You can delete an inbox only if it does not contain any issues, is not referenced in routing rules, and is not the default inbox of a user. Before you try to delete an inbox, edit the routing rules to prevent new issues from being routed to the inbox, and then reassign the existing issues in the inbox.

 **Note:** You need the **Groups, inboxes and user accounts** security privilege to add, edit, or delete inboxes.

To delete an inbox:

1. In the menu, click **Issues > Configuration**.
2. Click **Inboxes**. The **Inboxes** page opens.
3. In the **Actions** column of the inbox that you want to delete, click . A confirmation dialog box opens.
4. Click **Yes** to remove the inbox from the system.

Managing Project-Wide Notifications

Describes how to manage project-wide notifications.



Note: You need the **Manage email notifications** security privilege to add, edit, or delete project-wide notifications.

Adding Project-Wide Notifications

Describes how to add a project-wide notification.

1. Click **Issues > Configuration > Notifications**. All existing project-wide notifications are listed here.
2. Click **Add Notification**.
3. On the **Add Notification** dialog box, select a rule name from the list.
4. Select a radio button in the **From** area of the dialog box to indicate whether the return address should be:
 - The user who edits the issue (thereby setting off the notification).
 - Other user (select a user from the list).
5. Type the email addresses of the recipient(s) in the **To** field, separating multiple entries with commas. You may select users from the **Add Address** list .



Note: Although you can enter any email address in the **To** field, only users who have email addresses specified in their user accounts will appear in the **Add Address** list. For this reason it is recommended that email addresses be defined for all user accounts.

6. You can fill in the optional message text fields:
 - **Subject** is an optional subject line (up to 80 characters).
 - **Introduction** is optional introductory text (up to 250 characters)
7. Choose the content type of the message:
 - Contents can be:
 - **Issue Summary:** The information on the top half of the Issue Details page.
 - **Issue Summary with Description:** Issue summary plus the information from the Description tab.
 - **Full Issue Detail:** All issue information (this is the default).
 - Trailer is an optional signature field (up to 250 characters long).
8. Click **OK** to save the notification and to close the dialog.
 - To edit the notification, click on the **Rule Name**.
 - To delete the notification click  (**Delete Notification**) in the **Actions** column.
 - To create a copy of the notification, click .



Note: You need the security privilege **Manage email notifications** to define project-wide notifications.

Editing Project-Wide Notifications

Use the **Notifications** tab to edit existing project-wide notifications.

1. Click **Issues > Configuration > Notifications**. All existing project-wide notifications are listed here.
2. Click the **Edit** icon in the **Actions** column of the notification you want to edit.
3. On the **Add Notification** dialog box, select a rule name from the list.
4. Select a radio button in the **From** area of the dialog box to indicate whether the return address should be:
 - The user who edits the issue (thereby setting off the notification).
 - Other user (select a user from the list).

5. Type the email addresses of the recipient(s) in the **To** field, separating multiple entries with commas. You may select users from the **Add Address** list .



Note: Although you can enter any email address in the **To** field, only users who have email addresses specified in their user accounts will appear in the **Add Address** list. For this reason it is recommended that email addresses be defined for all user accounts.

6. You can fill in the optional message text fields:
 - **Subject** is an optional subject line (up to 80 characters).
 - **Introduction** is optional introductory text (up to 250 characters)
7. Choose the content type of the message:
 - Contents can be:
 - **Issue Summary:** The information on the top half of the Issue Details page.
 - **Issue Summary with Description:** Issue summary plus the information from the Description tab.
 - **Full Issue Detail:** All issue information (this is the default).
 - Trailer is an optional signature field (up to 250 characters long).
8. Click **OK** to save the notification and to close the dialog.

Deleting Project-Wide Notifications

1. Click **Issues > Configuration > Notifications**. All existing project-wide notifications are listed here.
2. Click the **Delete** icon in the **Actions** column of the notification you wish to delete.
3. Click **Yes** on the confirmation dialog box to execute the deletion.

Entering Parameters in Notification Subject

You can enter parameters in the subject field of a notification to generate a dynamic subject for email notifications.

The following parameters can be entered in the subject field:

| | |
|---------------------|---|
| {ID} | Returns the defect number |
| {PROJ} | Returns the project name |
| {PROD} | Returns the product |
| {COMP} | Returns the component |
| {USER} | Returns the user who modified the issue |
| {RULE} | Returns the name of the rule that triggered the notification |
| {SYN} | Returns the synopsis |
| {STATE} | Returns the current state of the issue |
| {SEV} | Returns the severity |
| {REASON} | Returns the current reason code of the issue |
| {CUSTOM1-40} | Returns the custom field of the specified number, e.g., {CUSTOM3} returns the value stored in custom field 3. |

For example, if you want to create an email notification that informs you whenever the state of an issue changes, you could add the following text into the subject field:

```
State of issue {ID} was changed by {user}.
```

If Bill would fix the currently open issue no. 2, you would receive an email notification with the following subject:

```
State of issue 2 was changed by bpetersson.
```

Enabling Basic Email Notification

A basic notification setting that can be activated by any user, regardless of assigned permissions, allows users to receive a notification when another user makes a change to an issue in their inbox or assigns a new issue to them.



Note: The subject line and content of these notifications can be configured in the file `SRFrontendBootConf.xml`.

1. Click **Configuration > Preferences**.
2. Select the **Notify me when users assign issues to me or make changes to issues in my inbox** checkbox.

Displaying Project-Wide Notifications

When a user selects the **Show project-wide notifications on each issue's Notifications dialog** check box (**Issues > Configuration > Preferences**), all project-wide notifications are listed on each issue's **Notification** dialog. Any notifications that are assigned to the active issue are also listed.

1. Click **Issues > Configuration > Preferences**.
2. Select the **Show project-wide notifications on each issue's Notifications dialog** check box.

Reports

The Reporting Data Mart

Silk Central offers a data mart for easy access to the most important data. This makes the creation of reports easier and improves the performance due to much simpler SQL queries.

Overview

The Silk Central reporting data mart makes it easy to access data for reporting purposes. It moves data from the production tables into dedicated views which should be used for creating advanced reports. The advantages include:

- Clear naming of tables and views, allowing you to quickly locate the data you are looking for.
- Pre-processed data, giving you the possibility to access aggregated data without having to calculate it yourself.
- Performance improvement, as reports can use much simpler and faster SQL queries.
- Less dependency on production database load, which also improves performance and removes load spikes.

The current version of the data mart covers the results area. Further areas for reporting will be added to the data mart in future releases. The following tables and views are currently available:

- The `DM_TestStatus` table is the basis for status-related views.
- The `RV_TestStatusExtended` view provides detailed information for a certain test execution.
- The `RV_LatestTestStatus` view provides status and extended information on the latest test run of a test within the context of a specific execution plan and build.
- The `RV_LatestFinishedTestStatus` view provides status and extended information on the latest test run of a test within the context of a specific execution plan and build. This view provides similar information as the `RV_LatestTestStatus` view, but only for test runs within finished execution plan runs.
- The `RV_MaxTestRunID` view is a helper to retrieve the latest test run ID for every test, execution plan, and build combination.

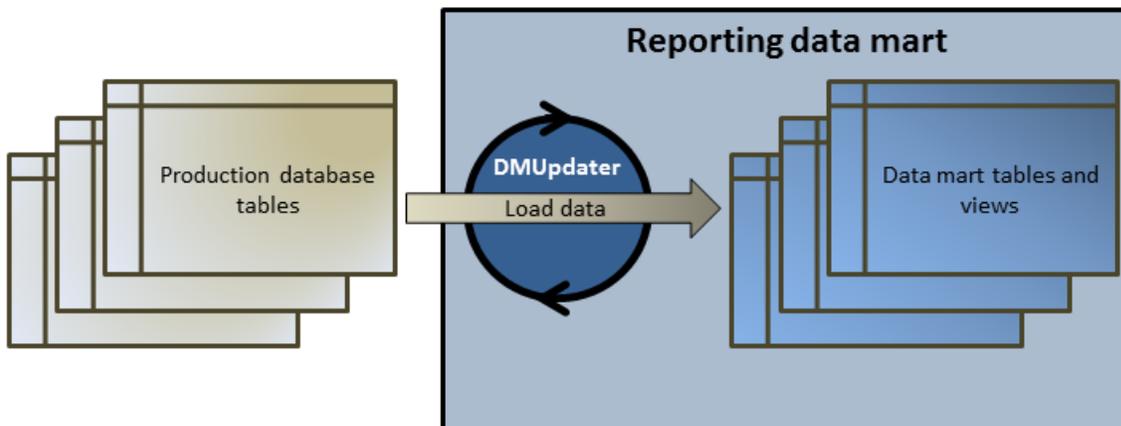
- The `RV_MaxFinishedTestRunID` view is a helper to retrieve the latest test run ID for every test, execution plan, and build combination. This view provides similar information as the `RV_MaxTestRunID` view, but only for test runs from finished execution plan runs.
- The `RV_TestingCycleStatus` view provides status information for testing cycles.
- The `RV_ExecutionPlanStatusPerBuild` view retrieves the latest test status sums for every execution plan in context of builds.
- The `RV_EPFinishedStatusPerBuild` view retrieves the latest test status sums for every execution plan in context of builds. This view provides similar information as the `RV_ExecutionPlanStatusPerBuild` view, but only for test runs from finished execution plan runs.
- The `RV_ExecutionPlanStatusRollup` view retrieves the sums for passed, failed, and not-executed tests per execution plan or folder in context of a build.
- The `RV_EPFinishedStatusRollup` view retrieves the sums for passed, failed, and not-executed tests per execution plan or folder in context of a build. This view provides similar information as the `RV_ExecutionPlanStatusRollup` view, but only for test runs from finished execution plan runs.
- The `RV_ConfigurationSuiteStatus` view lists the status counts for all configuration suites and configurations per build.
- The `RV_ConfigSuiteFinishedStatus` view lists the status counts for all configuration suites and configurations per build. This view provides similar information as the `RV_ConfigurationSuiteStatus` view, but only for test runs from finished execution plan runs.

You can download a .zip file with detailed information about the database schema of Silk Central. In the menu, click **Help** > **Documentation** and then click **Silk Central Database Schema** to download the .zip file.

Architecture

Data is periodically extracted in the background from the production database tables and loaded into the data mart tables and views for easy and fast querying. If the load on the database is not too high, this data is usually available within less than a minute after any changes have been committed. If you are logged in as a system administrator, you can check the current state of the data loading process by navigating to `http://<server>:<port>/sctm/check/db` and checking the **DM_TestStatus Table**.

 **Note:** If you are updating from a Silk Central version that did not include the data mart (before version 13.0), the data mart tables and views are initially filled with data from the production system. Depending on your database size, this process can take some time. Once this process has completed, you can access the data.



How to Create Reports with the Data Mart

The following examples demonstrate how to create useful reports with the data mart views.

Writing Data Mart Queries

1. In the menu, click **Reports > Details View**.
2. In the **Reports** tree, select the folder in which you want the new report to display.
This determines where the report is stored in the directory tree.
3. Click  on the toolbar. The **Create New Report** dialog box opens.
4. Type the name of the new report.
This is the name that is displayed in the **Reports** tree.
5. Check the **Share this report with other users** check box if you want to make this report available to other users.
6. Type a description of the report in the **Description** field.
7. Click **Advanced Query** to open the **Report data query** field. Insert previously written code or write new code directly in the field.
The **Insert placeholder** list assists you in editing the SQL queries with predefined function placeholders. For details, see *SQL Functions for Custom Reports*.
 **Note:** If you manually edit SQL code for the query, click **Check SQL** to confirm your work.
8. Click **Finish** to save your settings.

Reliability of Tests in an Execution Plan

Problem

In a continuous integration environment tests are ideally executed at least once per day for testing the daily build and ensuring the quality of your application under test. To understand how reliable your test set is for measuring the quality of your AUT it is inevitable to sometimes have a look at how the results change over time. For example you might have tests in your test set that frequently change status, therefore being no real measure for quality.

Solution

Use the data mart view [RV_TestStatusExtended](#) to create a report that lists the results for a specific test in the context of a specific execution plan. This allows you to see how this test's results have changed over time. For convenience, we will narrow the list of results down to those related to tagged builds, thus looking at specific milestone builds of the application under test only. This report collects test result data for tests in the context of execution plans and builds. In the following query we:

- Select the columns we want to display from this view.
- Narrow the result down by the ID of the test we want to investigate and the ID of the execution plan in which the test belongs.
- Add a constraint to consider tagged builds only.

```
SELECT TestName, ExecutionPlanName, VersionName, BuildName, TestRunID,
       PassedCount, FailedCount, NotExecutedCount
FROM RV_TestStatusExtended
WHERE TestID = ${TESTID|1|Test ID} AND ExecutionPlanID = ${EXECUTIONPLANID|1|
Execution Plan ID} AND BuildIsTagged = 1
ORDER BY BuildOrderNumber
```

The result of the SQL query are all test runs for the selected test within the selected execution plan. In the following example you can see that the test was re-run against build 579_Drop2:

| TestName | Execution PlanName | Version Name | BuildName | TestRunID | Passed Count | FailedCount | NotExecutedCount |
|----------|--------------------|--------------|------------|-----------|--------------|-------------|------------------|
| UI Tests | EN SQL2012 IE9 IIS | 3.0 | 579_Drop02 | 7741797 | 59 | 5 | 0 |
| UI Tests | EN SQL2012 IE9 IIS | 3.0 | 579_Drop02 | 7745078 | 63 | 1 | 0 |
| UI Tests | EN SQL2012 IE9 IIS | 3.0 | 593_Drop03 | 7787437 | 63 | 1 | 0 |
| UI Tests | EN SQL2012 IE9 IIS | 3.0 | 605_Drop04 | 7848720 | 63 | 1 | 0 |

All Failed Tests in an Execution Folder

Problem

Typically all execution plans are structured in a folder hierarchy which identifies the different areas or purposes to which the execution plans and their tests are related. The execution plans are triggered on a regular basis in a continuous integration environment, or occasionally over the release time frame, resulting in nice execution statistics – unfortunately for each single execution plan only.

However, sometimes you need an overall information to know how all your tests perform for a specific area or purpose to identify where the weaknesses are.

Solution

Use the data mart view [RV_LatestTestStatus](#) to create a report that returns a list of all failed tests for a specific execution planning hierarchy level.

The following query selects failed tests within an execution planning folder with context information like execution plan name and build name:

```
SELECT TestID, TestName, ExecutionPlanID, ExecutionPlanName, BuildName
FROM RV_LatestTestStatus lts
INNER JOIN TM_ExecTreePaths ON lts.ExecutionPlanID =
TM_ExecTreePaths.NodeID_pk_fk
WHERE TM_ExecTreePaths.ParentNodeID_pk_fk = ${executionFolderID|2179|
Execution Folder ID}
AND StatusID = 2
ORDER BY TestName
```

The query does the following:

- Uses the view [RV_LatestTestStatus](#) for retrieving the latest test run result.
- Includes the execution tree hierarchy ([TM_ExecTreePaths](#)) to be able to query all tests from all the execution plans within the hierarchy.
- Uses the top level folder ID from where the analysis should be started as [ParentNodeID_pk_fk](#).
- Includes only failed tests ([StatusID](#) = 2).

The [StatusID](#) can be looked up in the table [TM_TestDefStatusNames](#).

The result of the SQL query are all tests in the selected execution folder for which the last run failed.

| TestID | TestName | ExecutionPlanID | ExecutionPlanName | BuildName |
|--------|------------------|-----------------|-------------------|-----------|
| 14073 | JUnitTestPackage | 2184 | CI Testing | 352 |
| 14107 | Volatile Tests | 2191 | Volatile Test | 352 |

Testing Cycle Status

Problem

Testing cycles can be complex objects as they contain information about manual testers, tests, different builds and versions of products, and maybe even configurations. To not lose track it is important to find answers to questions like:

- How many tests have been finished?
- How many of them passed or failed per build?
- Are my manual testers still busy or can they do additional work?

Solution

Use the data mart view [RV_TestingCycleStatus](#) to create a report that shows the status of a testing cycle per tester and build that will give you an overview of how many tests are passed, failed, not executed grouped by manual tester, configuration and build.

```
SELECT BuildName, TesterLogin, TesterExecutionName,
       PassedCount, FailedCount, NotExecutedCount
FROM RV_TestingCycleStatus
WHERE TestingCycleID = ${testingCycleID|3|Testing Cycle ID}
ORDER BY BuildOrderNumber, TesterLogin
```

The query does the following:

- Uses the view [RV_TestingCycleStatus](#) as data source, as it contains `BuildName`, `TesterLogin` and `TesterExecutionName`, which is the generated name reflecting tester, configuration and test.
- Limits the data to the testing cycle ID that you are interested in.

The result of the SQL query shows the status per build and tester.

| BuildName | TesterLogin | TesterExecution Name | PassedCount | FailedCount | NotExecuted Count |
|-----------|--------------|---|-------------|-------------|-------------------|
| 352 | | No specific tester (Test Assets) | 0 | 0 | 1 |
| 351 | admin | admin (English SQL2008 FF Tomcat - Test Assets) | 0 | 1 | 0 |
| 352 | admin | admin (English SQL2008 FF Tomcat - Test Assets) | 0 | 0 | 1 |
| 352 | gmazzuchelli | gmazzuchelli (English Oracle10g IE8 Tomcat - Test Assets) | 0 | 1 | 1 |

| BuildName | TesterLogin | TesterExecution Name | PassedCount | FailedCount | NotExecuted Count |
|-----------|-------------|---|-------------|-------------|-------------------|
| 352 | jallen | jallen (German Oracle11g FF Tomcat - Test Assets) | 1 | 1 | 0 |
| 352 | smiller | smiller (German SQL2008 IE8 IIS - Test Assets) | 1 | 1 | 0 |

For unassigned tests a "no specific tester" group is created with empty values for TesterLogin, TesterFirstName, and TesterLastName.

In case you just want to see how your test cycle is doing based on the performance of your manual testers, a slight variation of the query will help:

```
SELECT TesterLogin, TesterExecutionName, SUM(PassedCount) PassedCount,
SUM(FailedCount) FailedCount, SUM(NotExecutedCount) NotExecutedCount
FROM RV_TestingCycleStatus
WHERE TestingCycleID = ${testingCycleID|3|Testing Cycle ID}
GROUP BY TesterLogin, TesterExecutionName
ORDER BY TesterLogin
```

The query is extended by:

- GROUP BY TesterLogin, TesterExecutionName for denoting the remaining columns.
- SUM() to the counters for aggregating the figures.

| TesterLogin | TesterExecution Name | PassedCount | FailedCount | NotExecutedCount |
|--------------|--|-------------|-------------|------------------|
| | No specific tester (Test Assets) | 0 | 0 | 1 |
| admin | admin (English SQL2008 FF Tomcat - Test Assets) | 0 | 1 | 1 |
| gmazzuchelli | gmazzuchelli (English Oracle10g IE8 Tomcat - Test Assets) | 0 | 1 | 1 |
| jallen | jallen (German Oracle11g FF Tomcat - Test Assets) | 1 | 1 | 0 |
| smiller | smiller (German SQL2008 IE8 IIS - Test Assets) | 1 | 1 | 0 |

Execution Tree Status

Problem

It is a common practice to have execution plans in a hierarchical structure that represents different testing areas or purposes. In some cases, for example for knowing the test status and therefore the quality of an area or purpose, you will want to know the overall passed, failed, and not executed count.

Solution

Use the data mart view [RV_ExecutionPlanStatusRollup](#) to create a report that returns the passed, failed, and not executed counts grouped by build for a specific execution planning folder.

```
SELECT BuildName, PassedCount, FailedCount, NotExecutedCount
FROM RV_ExecutionPlanStatusRollup
WHERE ExecutionFolderID = ${executionPlanID|43|Execution Plan ID}
```

The query does the following:

- Selects BuildName and the status counts from the RV_ExecutionPlanStatusRollup view.
- Specifies the top-level folder you want the status from (ExecutionFolderID).

The result of the SQL query shows the status of your test runs in all execution plans of the selected folder, aggregated per build.

| BuildName | PassedCount | FailedCount | NotExecutedCount |
|-----------|-------------|-------------|------------------|
| 351 | 0 | 0 | 2 |
| 352 | 15 | 7 | 1 |

If you are interested in more details, for example the status counts for each execution plan within the selected hierarchy, you can use the data mart view [RV_ExecutionPlanStatusPerBuild](#):

```
SELECT eps.BuildName, eps.ExecutionPlanID, SUM(eps.PassedCount) PassedCount,
SUM(eps.FailedCount) FailedCount, SUM(eps.NotExecutedCount) NotExecutedCount
FROM RV_ExecutionPlanStatusPerBuild eps
INNER JOIN TM_ExecTreePaths etp ON eps.ExecutionPlanID = etp.NodeID_pk_fk
WHERE etp.ParentNodeID_pk_fk = ${execFolderID|44|Execution Folder ID}
GROUP BY eps.ExecutionPlanID, eps.BuildOrderNumber, eps.BuildName
ORDER BY eps.BuildOrderNumber, eps.ExecutionPlanID
```

The query does the following:

- Uses the RV_ExecutionPlanStatusPerBuild view to access execution-plan specific data (ExecutionPlanID and ExecutionPlanName). The previously used RV_ExecutionPlanStatusRollup view contains pre-aggregated data (summed up data), which is not suitable for the purpose here as you would get results not only for execution plans but for the folder nodes as well.
- Selects all nodes within a specific folder with a JOIN of the TM_ExecTreePath table to bring in hierarchy information.
- Specifies the top-level folder with ExecutionFolderID. As the table TM_ExecutionTreePaths also contains a self-reference for every execution plan, you could run this query with an execution plan ID for ParentNodeID_pk_fk too, which would return the rows for the specific execution plan.
- Adds ORDER BY BuildOrderNumber and ExecutionPlanID to get a nicely ordered result, showing the oldest builds and their execution plans first.

The result of the SQL query shows the status of your test runs in all execution plans of the selected folder.

| BuildName | ExecutionPlanID | PassedCount | FailedCount | NotExecutedCount |
|-----------|-----------------|-------------|-------------|------------------|
| 351 | 2307 | 0 | 0 | 2 |
| 352 | 2184 | 11 | 2 | 0 |
| 352 | 2185 | 0 | 3 | 0 |
| 352 | 2186 | 2 | 1 | 0 |
| 352 | 2187 | 1 | 0 | 0 |
| 352 | 2191 | 0 | 1 | 0 |

| BuildName | ExecutionPlanID | PassedCount | FailedCount | NotExecutedCount |
|-----------|-----------------|-------------|-------------|------------------|
| 352 | 2307 | 1 | 0 | 1 |

Configuration Suite Status

Problem

Configuration suites allow you to execute the same set of tests against multiple configurations, for example multiple browsers or operating systems. To be able to make reasonable statements related to quality and reliability of your application under test you will want to keep track of the results for each individual configuration.

Solution

Use the data mart view [RV_ConfigurationSuiteStatus](#) to create a report that returns the passed, failed, and not executed counts for each configuration per build.

```
SELECT BuildName, ConfigurationName, PassedCount, FailedCount,
NotExecutedCount
FROM RV_ConfigurationSuiteStatus
WHERE ConfigurationSuiteID = ${configSuiteID|97|Configuration Suite ID}
ORDER BY BuildOrderNumber, ConfigurationName
```

The query does the following:

- Retrieves the status counts per build of test runs from the [RV_ConfigurationSuiteStatus](#) view.
- Narrows the results down to the configuration suite (ConfigurationSuiteID).

The result of the SQL query shows the status of your test runs for each configuration.

| BuildName | ConfigurationName | PassedCount | FailedCount | NotExecutedCount |
|-----------|-------------------|-------------|-------------|------------------|
| 350 | Chrome | 0 | 1 | 0 |
| 350 | Firefox | 0 | 1 | 0 |
| 350 | Internet Explorer | 0 | 1 | 0 |
| 351 | Chrome | 1 | 0 | 0 |
| 351 | Firefox | 1 | 0 | 0 |
| 351 | Internet Explorer | 0 | 1 | 0 |
| 352 | Chrome | 1 | 0 | 0 |
| 352 | Firefox | 1 | 0 | 0 |
| 352 | Internet Explorer | 1 | 0 | 0 |

In this example, we use the ID of the configuration suite to get all configurations. It is also possible to restrict the result to specific builds, in which case you would have to include `BuildID`, `BuildName`, or `BuildOrderNumber` in the where clause.



Note: The view [RV_ConfigurationSuiteStatus](#) only contains aggregated status counts without any test-specific data. To retrieve additional test-specific data, you can use, for example, the view [RV_LatestTestStatus](#).

Troubleshooting

Wrong or Missing Data

Problem

When querying data from a data mart table or view, the listed results are not up to date or missing.

Resolution

The data mart tables and views are updated periodically in the background, but not in real time. Due to this, it can take a few seconds up to a few minutes for the data to be loaded into the data mart tables. If your system is running a heavy load, this influences the performance of the background process which is loading the data. The reason is that other processes are prioritized higher and may temporarily block the DataMartUpdater background job. Run your query again later to retrieve updated data.

If you are logged in as a system administrator, you can check the current state of the data loading process by navigating to `http://<server>:<port>/sctm/check/db` and checking the **DM_TestStatus Table**.



Note: Tests and depending test runs are removed from the data mart if a test is deleted. This also applies to deleted tests due to cleaning up test packages.

The Data Mart Slows Down the System

Problem

Since running the data mart, the system's overall performance seems to be poorer or behaves inconsistently.

Resolution

While this should not happen, you can turn off the data mart to check if this actually resolves your issues:

1. On the **Instance Administration** page, stop the application server of the instance that you want to modify.
2. Open the `TMApServerHomeConf.xml` file with a text editor. The default path for this file is `C:\Program Files (x86)\Silk\Silk Central 20.6\instance_<instance number>_<instance name>\Conf\AppServer` on the application server.
3. Locate the `Config/DataMart/Enabled` XML tag and set the value to `false`.
4. Save and close the XML file.
5. Restart the application server.

Reference: Data Mart Tables and Views

The following data mart tables and views are available for easy and fast reporting.

DM_TestStatus

The `DM_TestStatus` table is the basis for status-related views.

The other data mart views usually provide easier access to detailed data, as this table does not provide direct access to information like the name of a test. The key of this table is the combination of the columns `TestID`, `ExecutionPlanID`, `BuildID`, and `TestRunID`.

| Row | Description |
|-----------------|-----------------------------------|
| TestID | Identifier of the test. |
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| TestRunID | Identifier of the test run. |

| Row | Description |
|----------------------------|--|
| ExecutionRunID | Identifies in which execution run this result was generated. |
| StatusID | Status of this test run (see TM_TestDefStatusNames). |
| EDRStatusID | Status of this execution run. For additional information, see TM_TestDefStatusNames . For example, status 7 = pending manual run, status 10 = pending automated run. |
| ReasonID | Reason for the status of this test run (see TM_ResultStatusReasons). Can be null. |
| PassedCount | Sum of all passed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| FailedCount | Sum of all failed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| NotExecutedCount | Sum of all not-executed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| ProjectID | ID of the project that this row belongs to. |
| TestStartTime | Time when the test run started (UTC). |
| ExecutionStartTime | Time when the execution run started (UTC). |
| TestDurationInMilliseconds | Duration of the test run in milliseconds. |
| IsBlocked | Flags the test run as blocked/unblocked |
| DbChangedAt | Time when this row was last updated by the reporting data mart. |

RV_TestStatusExtended

The `RV_TestStatusExtended` view provides detailed information for a certain test execution.

This view contains all test runs, in contrast to the view [RV_LatestTestStatus](#) which contains only the latest test run of a test within the context of an execution plan and a certain build. You can use this view for example to create a [report that lists all test runs of your tagged builds](#). The key of this table is the combination of the columns `TestID`, `ExecutionPlanID`, `BuildID`, and `TestRunID`.

 **Note:** Tests and depending test runs are removed from the data mart if a test is deleted. This also applies to deleted tests due to cleaning up test packages.

| Row | Description |
|-----------------|---|
| TestID | Identifier of the test. |
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| TestRunID | Identifier of the test run. |
| ExecutionRunID | Identifies in which execution run this result was generated. |
| StatusID | Status of this test run (see TM_TestDefStatusNames). |

| Row | Description |
|----------------------------|---|
| ReasonID | Reason for the status of this test run (see TM_ResultStatusReasons). Can be null. |
| PassedCount | Sum of all passed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| FailedCount | Sum of all failed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| NotExecutedCount | Sum of all not-executed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| ProjectID | ID of the project that this row belongs to. |
| TestStartTime | Time when the test run started (UTC). |
| ExecutionStartTime | Time when the execution run started (UTC). |
| TestDurationInMilliseconds | Duration of the test run in milliseconds. |
| IsBlocked | Flags the test run as blocked/unblocked |
| DbChangedAt | Time when this row was last updated by the reporting data mart. |
| TestName | Name of the test. |
| TestDescription | Description of the test. |
| TestParentID | ID of the test's parent. |
| PlannedTimeInMinutes | Time planned for this test in minutes. |
| Reason | Name of the reason. Can contain reasons that have been deleted in the meantime. |
| ExecutionPlanName | Name of the execution plan. |
| ExecutionPlanDescription | Description of the execution plan. |
| ExecutionParentFolderID | ID of the execution plan's parent. |
| Priority | Priority of the execution plan: 0 = Low, 1 = Medium, 2 = High. |
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |

| Row | Description |
|--------------------|------------------------------|
| ProductOrderNumber | Order number of the product. |

RV_LatestTestStatus

The `RV_LatestTestStatus` view provides status and extended information on the latest test run of a test within the context of a specific execution plan and build.

Use the `RV_TestStatusExtended` view to retrieve information about all test runs. You can use this view to create a [report that lists all failed tests in an execution folder](#). The key of this table is the combination of the columns `TestID`, `ExecutionPlanID`, `BuildID`, and `TestRunID`.

| Row | Description |
|----------------------------|---|
| TestID | Identifier of the test. |
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| TestRunID | Identifier of the test run. |
| ExecutionRunID | Identifies in which execution run this result was generated. |
| StatusID | Status of this test run (see <code>TM_TestDefStatusNames</code>). |
| ReasonID | Reason for the status of this test run (see <code>TM_ResultStatusReasons</code>). Can be null. |
| PassedCount | Sum of all passed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| FailedCount | Sum of all failed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| NotExecutedCount | Sum of all not-executed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| ProjectID | ID of the project that this row belongs to. |
| TestStartTime | Time when the test run started (UTC). |
| ExecutionStartTime | Time when the execution run started (UTC). |
| TestDurationInMilliseconds | Duration of the test run in milliseconds. |
| IsBlocked | Flags the test run as blocked/unblocked |
| DbChangedAt | Time when this row was last updated by the reporting data mart. |
| TestName | Name of the test. |
| TestDescription | Description of the test. |
| TestParentID | ID of the test's parent. |
| PlannedTimeInMinutes | Time planned for this test in minutes. |
| Reason | Name of the reason. Can contain reasons that have been deleted in the meantime. |
| ExecutionPlanName | Name of the execution plan. |

| Row | Description |
|--------------------------|--|
| ExecutionPlanDescription | Description of the execution plan. |
| ExecutionParentFolderID | ID of the execution plan's parent. |
| Priority | Priority of the execution plan: 0 = Low, 1 = Medium, 2 = High. |
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

RV_LatestFinishedTestStatus

The `RV_LatestFinishedTestStatus` view provides status and extended information on the latest test run of a test within the context of a specific execution plan and build. This view provides similar information as the `RV_LatestTestStatus` view, but only for test runs within finished execution plan runs.

Use this view to create reports that ignore currently running execution plans.

Use the [RV_TestStatusExtended](#) view to retrieve information about all test runs. The key of this table is the combination of the columns `TestID`, `ExecutionPlanID`, `BuildID`, and `TestRunID`.

| Row | Description |
|-----------------|---|
| TestID | Identifier of the test. |
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| TestRunID | Identifier of the test run. |
| ExecutionRunID | Identifies in which execution run this result was generated. |
| StatusID | Status of this test run. For additional information, see the <code>TM_TestDefStatusNames</code> table in the database schema of Silk Central. |
| EDRStatusID | Status of this execution run. For example, status 7 = pending manual run, status 10 = pending automated run. For additional information, see the <code>TM_TestDefStatusNames</code> table in the database schema of Silk Central. |

| Row | Description |
|----------------------------|---|
| ReasonID | Reason for the status of this test run. Can be null. For additional information, see the TM_ResultStatusReasons table in the database schema of Silk Central. |
| PassedCount | Sum of all passed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| FailedCount | Sum of all failed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| NotExecutedCount | Sum of all not-executed tests, which is 0 or 1 for common tests and can be more for package test roots. |
| ProjectID | ID of the project that this row belongs to. |
| TestStartTime | Time when the test run started (UTC). |
| ExecutionStartTime | Time when the execution run started (UTC). |
| TestDurationInMilliseconds | Duration of the test run in milliseconds. |
| IsBlocked | Flags the test run as blocked/unblocked |
| DbChangedAt | Time when this row was last updated by the reporting data mart. |
| TestName | Name of the test. |
| TestDescription | Description of the test. |
| TestParentID | ID of the test's parent. |
| PlannedTimeInMinutes | Time planned for this test in minutes. |
| Reason | Name of the reason. Can contain reasons that have been deleted in the meantime. |
| ExecutionPlanName | Name of the execution plan. |
| ExecutionPlanDescription | Description of the execution plan. |
| ExecutionParentFolderID | ID of the execution plan's parent. |
| Priority | Priority of the execution plan: 0 = Low, 1 = Medium, 2 = High. |
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

RV_MaxTestRunID

The RV_MaxTestRunID view is a helper to retrieve the latest test run ID for every test, execution plan, and build combination.

The key of this table is the combination of the columns TestID, ExecutionPlanID and BuildID.

| Row | Description |
|-----------------|---|
| TestID | Identifier of the test. |
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| MaxTestRunID | Identifies the latest test run for the test in context of the execution plan and build. |

RV_MaxFinishedTestRunID

The RV_MaxFinishedTestRunID view is a helper to retrieve the latest test run ID for every test, execution plan, and build combination. This view provides similar information as the RV_MaxTestRunID view, but only for test runs from finished execution plan runs.

Use this view to create reports that ignore currently running execution plans.

The key of this table is the combination of the columns TestID, ExecutionPlanID, and BuildID.

| Row | Description |
|-----------------|---|
| TestID | Identifier of the test. |
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| MaxTestRunID | Identifies the latest test run for the test in context of the execution plan and build. |

RV_TestingCycleStatus

The RV_TestingCycleStatus view provides status information for testing cycles.

You can use this view to create a [report that shows the current status of a testing cycle](#).

TestingCycleID denotes the testing cycle and TesterExecutionID (as well as TesterExecutionName, UserID, CapacityInCycle, TesterLogin, TesterFirstName, TesterLastName) is used to identify the assigned tester in the testing cycle. For the tests which are not assigned to a specific tester, the UserID, CapacityInCycle, TesterLogin, TesterFirstName, and TesterLastName are null. The key of this table is the combination of the columns TesterExecutionID and BuildID.

| Row | Description |
|--------------------------|---|
| TestingCycleID | Identifier of the testing cycle. |
| TesterExecutionID | Identifies the group of tests that are assigned to a specific tester. |
| TesterExecutionName | The generated name for the group of tests that are assigned to a specific tester. |
| UserID | The user ID of the tester. |
| CapacityInCycleInMinutes | The capacity for this user in this testing cycle in minutes. |
| TesterLogin | Login name of the tester. |

| Row | Description |
|--------------------|--|
| TesterFirstName | First name of tester. |
| TesterLastName | Last name of tester. |
| PassedCount | Sum of all passed tests. |
| FailedCount | Sum of all failed tests. |
| NotExecutedCount | Sum of all not-executed tests. |
| ProjectID | Identifier of the project. |
| BuildID | Identifier of the build. |
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

RV_ExecutionPlanStatusPerBuild

The RV_ExecutionPlanStatusPerBuild view retrieves the latest test status sums for every execution plan in context of builds.

Folders and child nodes are not considered. You can use this view to create a [report that shows the status of your test runs for each execution plan in a folder](#). In contrast to [RV_ExecutionPlanStatusRollup](#), this view has a slight performance advantage as no hierarchy is considered for retrieving the data. The key of this table is the combination of the columns ExecutionPlanID and BuildID.

| Row | Description |
|-------------------------|---|
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| ExecutionPlanName | Name of the execution plan. |
| ExecutionParentFolderID | ID of the execution plan's parent. |
| PassedCount | Sum of all passed tests. |
| FailedCount | Sum of all failed tests. |
| NotExecutedCount | Sum of all not-executed tests. |
| ProjectID | ID of the project that the execution plan belongs to. |

| Row | Description |
|--------------------|--|
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

RV_EPFinishedStatusPerBuild

The RV_EPFinishedStatusPerBuild view retrieves the latest test status sums for every execution plan in context of builds. This view provides similar information as the RV_ExecutionPlanStatusPerBuild view, but only for test runs from finished execution plan runs.

Use this view to create reports that ignore currently running execution plans. Folders and child nodes are not considered. You can use this view to create a [report that shows the status of your test runs for each execution plan in a folder](#). In contrast to *RV_ExecutionPlanStatusRollup*, this view has a slight performance advantage as no hierarchy is considered for retrieving the data. The key of this table is the combination of the columns `ExecutionPlanID` and `BuildID`.

| Row | Description |
|-------------------------|---|
| ExecutionPlanID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| ExecutionPlanName | Name of the execution plan. |
| ExecutionParentFolderID | ID of the execution plan's parent. |
| PassedCount | Sum of all passed tests. |
| FailedCount | Sum of all failed tests. |
| NotExecutedCount | Sum of all not-executed tests. |
| ProjectID | ID of the project that the execution plan belongs to. |
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |

| Row | Description |
|--------------------|--|
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

RV_ExecutionPlanStatusRollup

The RV_ExecutionPlanStatusRollup view retrieves the sums for passed, failed, and not-executed tests per execution plan or folder in context of a build.

In case of folders, the counters include the numbers from all children. You can use this view to create a [report that shows the status of all test runs in a folder](#). The key of this table is the combination of the columns ExecutionFolderID and BuildID.

| Row | Description |
|--------------------|---|
| ExecutionFolderID | Identifier of the execution plan. |
| BuildID | Identifier of the build. |
| PassedCount | Sum of all passed tests. |
| FailedCount | Sum of all failed tests. |
| NotExecutedCount | Sum of all not-executed tests. |
| ProjectID | ID of the project that the execution plan belongs to. |
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

RV_EPFInishedStatusRollup

The RV_EPFInishedStatusRollup view retrieves the sums for passed, failed, and not-executed tests per execution plan or folder in context of a build. This view provides similar information as the RV_ExecutionPlanStatusRollup view, but only for test runs from finished execution plan runs.

Use this view to create reports that ignore currently running execution plans.

In case of folders, the counters include the numbers from all children. The key of this table is the combination of the columns `ExecutionFolderID` and `BuildID`.

| Row | Description |
|---------------------------------|---|
| <code>ExecutionFolderID</code> | Identifier of the execution plan. |
| <code>BuildID</code> | Identifier of the build. |
| <code>PassedCount</code> | Sum of all passed tests. |
| <code>FailedCount</code> | Sum of all failed tests. |
| <code>NotExecutedCount</code> | Sum of all not-executed tests. |
| <code>ProjectID</code> | ID of the project that the execution plan belongs to. |
| <code>BuildName</code> | Name of the build used for this test run. |
| <code>BuildDescription</code> | Description of the build. |
| <code>BuildOrderNumber</code> | Order number of the build. |
| <code>BuildIsTagged</code> | 1 if the build is tagged, 0 otherwise. |
| <code>VersionID</code> | ID of the version that the build belongs to. |
| <code>VersionName</code> | Name of the version. |
| <code>VersionDescription</code> | Description of the version. |
| <code>VersionOrderNumber</code> | Order number of the version. |
| <code>ProductID</code> | ID of the product that the build belongs to. |
| <code>ProductCode</code> | Name of the product. |
| <code>ProductDescription</code> | Description of the product. |
| <code>ProductOrderNumber</code> | Order number of the product. |

RV_ConfigurationSuiteStatus

The `RV_ConfigurationSuiteStatus` view lists the status counts for all configuration suites and configurations per build.

You can use this view to create a [report that shows the status of all test runs for each configuration in a configuration suite](#). The key of this table is the combination of the columns `ConfigurationID` and `BuildID`.

| Row | Description |
|-----------------------------------|---|
| <code>ConfigurationSuiteID</code> | Identifier of the configuration suite. |
| <code>ConfigurationID</code> | Identifier of the configuration. |
| <code>ConfigurationName</code> | Name of the configuration. |
| <code>BuildID</code> | Identifier of the build. |
| <code>PassedCount</code> | Sum of all passed tests. |
| <code>FailedCount</code> | Sum of all failed tests. |
| <code>NotExecutedCount</code> | Sum of all not-executed tests. |
| <code>ProjectID</code> | ID of the project that this row belongs to. |

| Row | Description |
|--------------------|--|
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

RV_ConfigSuiteFinishedStatus

The RV_ConfigSuiteFinishedStatus view lists the status counts for all configuration suites and configurations per build. This view provides similar information as the RV_ConfigurationSuiteStatus view, but only for test runs from finished execution plan runs.

Use this view to create reports that ignore currently running execution plans.

The key of this table is the combination of the columns ConfigurationID and BuildID.

| Row | Description |
|----------------------|--|
| ConfigurationSuiteID | Identifier of the configuration suite. |
| ConfigurationID | Identifier of the configuration. |
| ConfigurationName | Name of the configuration. |
| BuildID | Identifier of the build. |
| PassedCount | Sum of all passed tests. |
| FailedCount | Sum of all failed tests. |
| NotExecutedCount | Sum of all not-executed tests. |
| ProjectID | ID of the project that this row belongs to. |
| BuildName | Name of the build used for this test run. |
| BuildDescription | Description of the build. |
| BuildOrderNumber | Order number of the build. |
| BuildIsTagged | 1 if the build is tagged, 0 otherwise. |
| VersionID | ID of the version that the build belongs to. |
| VersionName | Name of the version. |
| VersionDescription | Description of the version. |
| VersionOrderNumber | Order number of the version. |

| Row | Description |
|--------------------|--|
| ProductID | ID of the product that the build belongs to. |
| ProductCode | Name of the product. |
| ProductDescription | Description of the product. |
| ProductOrderNumber | Order number of the product. |

Reporting Area

Overview

The default reports that ship with Silk Central are visible in all projects and are listed under **Global Reports** in the **Reports** tree. These reports cannot be modified, but you can make copies that you can customize to your needs. Creating your own custom reports within the **Global Reports** tree makes your reports available for all projects too. You can also copy reports between project trees and the **Global Reports** tree. The concept of global reports allows you to flexibly administrate reports across all projects within Silk Central.

Limitations:

- Because global reports are not in context of a project, they need to be advanced reports (SQL).
- Global reports cannot be sub-reports of project reports, and sub-reports are not available in the global reports section.

Shared Global Report Permissions

Because shared global reports are available in all projects for all users, user roles that are able to update and delete them require special permissions: **Manage shared global reports** and **Delete shared global reports**. These permissions also apply to report templates that are used in shared global reports. Non-shared global reports and report templates can be updated and deleted by user roles with the standard report permissions for updating and deleting reports. This implies the following special scenarios:

- The option **Share this report with other users** is disabled in global reports and folders if the user does not have the **Manage shared global reports** permission
- When pasting a copied shared report into the **Global Reports** tree without the **Manage shared global reports** permission, the pasted report is set to non-shared

Creating and Customizing Reports

Creating Reports

Silk Central offers reports that quickly and easily transform data into intuitive charts and graphs. Pre-installed reports are available for the **Requirements**, **Tests**, and **Issues** areas.

Reports are created using either BIRT Report Designer, an open-source, Eclipse-based report tool, or Microsoft Word or Microsoft Excel report templates. Silk Central is tightly integrated with BIRT Report Designer to make it easy for you to generate reports on test management data. The reporting functionality in Silk Central is highly customizable. Numerous pre-installed reports and report templates provide out-of-the-box options for a wide range of reporting needs. Simple GUI-based tools allow you to edit the pre-installed reports and create reports of your own. For users with SQL knowledge, there is virtually no limit to how data can be queried and presented in custom reports.



Tip: If a blank report is generated, the cause may be that there are not any data in the project you selected, or you may not be connected to the appropriate Silk Central database. Reports are not available offline unless your Silk Central database is accessible locally.

Creating New Reports

To create a new report:

1. In the menu, click **Reports > Details View**.
2. In the **Reports** tree, select the folder in which you want the new report to display.
This determines where the report is stored in the directory tree.
3. Click  on the toolbar. The **Create New Report** dialog box opens.
4. Type the name of the new report.
This is the name that is displayed in the **Reports** tree.
5. Check the **Share this report with other users** check box if you want to make this report available to other users.
6. In the **Timeout [s]** field, type the maximum time period in seconds that Silk Central should wait for SQL queries to complete.
7. From the **Default tab** list, select the tab that you want to be directed to when you select this report from one of the context-sensitive report lists.
8. Select the corresponding result type from the **Result category** list.
This setting specifies the database table and view that is to be filtered for the report. The following result types are available:

| Result Type | Description |
|------------------------------------|---|
| Requirement | Returns requirements available in the Requirements module that meet the query criteria. |
| Test | Returns tests available in the Tests area that meet the query criteria. |
| Test Execution | Returns executed test results from the Executions area that meet the query criteria. |
| Execution Plan | Returns execution plans from the execution area. |
| Issue | Returns issues, including imported issues. |
| Requirement Progress Builds | Contains information on requirements progress per build so that you can see how requirements develop across builds. |
| Requirement Progress Days | The same as Requirement Progress Builds, but shows development on a daily basis. |
| Test Progress Builds | Shows how tests develop across builds. |
| Test Progress Days | Same as above, but shows development on a daily basis. |

Each result type offers a set of selection criteria. Based on the result type you have selected, specify an appropriate **Selection criteria** for your report. These criteria typically group properties based on a view or some other intuitive grouping, for example custom properties.

9. From the **Property** list, select the property that is to be filtered on.
For some selection criteria, properties are dynamic.
10. Select an **Operator** for the query.
The available operators depend on the property. Example operators are `=`, `not`, `like`, and `not like`. Strings are always compared lowercase. Allowed wildcards for strings are `"*` and `"?"`, where `*` matches any characters and `?` matches exactly one character.
11. Select or specify the **Value** that the query is to be filtered on.
For date-based properties, the **Value** field is replaced with a calendar tool that you can use to select a specific date.
12. *Optional:* To add an additional query string to this report, click **More**. An existing query string can be deleted by clicking the string's **Delete** button. When multiple query strings are defined, **AND** and **OR**

option buttons are displayed next to **More**. Use these option buttons to define if the queries should be considered cumulatively, or if only one query string's criteria needs to be met.

13. Click **Next** to configure report columns on the **New Report** dialog box.

14. Click **Add Columns**. The **Add Columns** dialog box lists all available report columns.

15. Select the columns that you want to have included in the report and click **OK**.

You can select multiple columns with **Ctrl+Click**.

 **Note:** For test-planning reports, the list of available column names is enhanced with the column names from the *LQM_V_Tests* table. For details, refer to the *Silk Central Database Model Schema*.

The selected columns display in tabular format on the **New Report** dialog box.

16. *Optional:* Configure how each report column is to be displayed. For each column, specify a sort direction, *ascending*, *descending*, or *unsorted*, using the up/down arrows in the **Sorting** column.

17. When multiple columns are selected for sorting, a list box is displayed in the **Sort Order** column that allows you to more easily edit the column-sort order. Set these numbers as required.

18. Give each column an **Alias**.

This is the name by which each column will be labeled in the generated report.

 **Note:** An **Alias** cannot contain double quotes (").

19. With grouping, you can take advantage of SQL aggregation features, for example when selecting a number of elements or querying a total sum of values. Check the **Group by** check box to specify that SQL group by functions are to be applied.

20. Columns that are not selected for SQL group by functions are set to aggregation by default, which means a single aggregate value is calculated. From the **Aggregation** list, select the appropriate aggregation type.

The following types are available:

- Count
- Sum
- Average
- Minimum
- Maximum

21. The **Actions** column enables you to move column listings up and down in the view, or to delete a column.

22. Click **Finish** to complete your new report.

SQL Functions for Custom Reports

To assist in writing advanced queries, placeholders are available for each function. Function placeholders are replaced with SQL code upon execution. Functions are used like parameters, but their names have a \$ (dollar symbol) as a prefix. Unlike parameters, placeholders are defined report elements that cannot be customized per execution.

The following table lists all available function placeholders:

| Function | What it does | Example |
|----------------|--|--|
| \$TODAY | Returns the current systemdate on the database server. You can also write \$TODAY-1 for yesterday or \$TODAY-7 for a week ago. | CreatedAt > \${\$TODAY} |
| \$DATE(column) | Returns the timestamp of the beginning of the day in UTC. | The column returns: 2019-08-30 08:37:33 in CEST, which is rendered as 2019-08-30 02:00 |

| Function | What it does | Example |
|------------------|--|---|
| \$DATE('string') | Converts the given string to a database date. | CreatedAt > \$ { \$DATE('01/10/2005') } |
| \$DAYS[p1;p2] | Calculates the difference in days between the two given parameters. The two parameters can be a column within the table/view or \$TODAY. | The following example returns the rows created within the last week: \$ { \$DAYS[CreatedAt ; \$TODAY] } < 7 |
| \$WEEK(param) | Returns the week-number of the given parameter, which can be \$TODAY or a column. | |
| \$MONTH(param) | Returns the month of the year as a number of the given parameter, which can be \$TODAY or a column. | |
| \$YEAR(param) | Returns the year as a number of the given parameter, which can be \$TODAY or a column. | |
| \$USERID | The ID of the currently logged in user. | |
| \$USERNAME | The name of the currently logged in user. | |
| \$PROJECTID | The ID of the currently selected project. | |
| \$PROJECTNAME | The name of the currently selected project. | |
| \$REPORTNAME | The name of the currently selected report. | |
| \$REPORTID | The ID of the currently selected report. | |

Below is the code of the pre-installed *Requirement with Child Requirements* report. With this report, a selected requirement is shown with its requirement ID. Full details regarding the requirement's child requirements are displayed. Although not a custom report, this report is a helpful example because it makes use of the \$PROJECTID function. It also includes two parameters, reqID (requirement ID) and reqProp_Obsolete_0 (show obsolete requirements).

```
SELECT r.ReqID, r.ReqCreated, r.ReqName, r.TreeOrder
FROM RTM_V_Requirements r INNER JOIN
TM_ReqTreePaths rtp ON (rtp.ReqNodeID_pk_fk = r.ReqID)
WHERE rtp.ParentNodeID_pk_fk=${reqID|22322|Requirement ID} AND
r.ProjectID = ${$PROJECTID} AND
r.MarkedAsObsolete=${reqProp_Obsolete_0|0|Show obsolete Requirements}
ORDER BY r.TreeOrder ASC
```

Writing Advanced Queries with SQL

Advanced reports can be created through manual SQL coding. Virtually any reporting option is available if you know the database schema. Clicking **Advanced Query** hides the query string list boxes and opens the **Report data query** field in which you can insert existing code or write new SQL code.

 **Restriction:** The SQL statement `select top` is not supported. It is not possible to report directly on tables which return user information from Silk Central, for example SCC_Users, SCC_AuditEntries, SCC_UserGroups, and SCC_UserGroupRoles.

One approach is to begin query-string construction using the list boxes as outlined in *Creating New Reports*. If the report criteria are valid, the equivalent SQL statement will be generated and displayed, and then move to advanced mode for further modifications.

 **Note:** If you switch from advanced mode back to simple mode the changes you made within the code will be lost.

To write an advanced query directly in SQL:

1. In the menu, click **Reports > Details View**.
2. In the **Reports** tree, select the folder in which you want the new report to display.
This determines where the report is stored in the directory tree.
3. Click  on the toolbar. The **Create New Report** dialog box opens.
4. Type the name of the new report.
This is the name that is displayed in the **Reports** tree.
5. Check the **Share this report with other users** check box if you want to make this report available to other users.
6. Type a description of the report in the **Description** field.
7. Click **Advanced Query** to open the **Report data query** field. Insert previously written code or write new code directly in the field.

The **Insert placeholder** list assists you in editing the SQL queries with predefined function placeholders. For details, see *SQL Functions for Custom Reports*.

 **Note:** If you manually edit SQL code for the query, click **Check SQL** to confirm your work.

8. Click **Finish** to save your settings.

Working with Subreports

Adding Subreports

To aggregate the results from multiple reports into the currently selected report, you can add subreports. When adding a report as a subreport, the result columns and rows of the subreport are concatenated to the results of the selected report.

To add a report as a subreport:

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Properties** tab.
4. Click **Add Subreport**.
The **Add Subreport** dialog box appears.
5. From the **Reports** tree, select the subreport you want to append to the current report.
6. Click **OK** to complete the addition of the subreport. Subreports are displayed on the associated report's **Properties** page in the **Subreports** section.

Deleting Subreports

To delete a subreport:

1. In the menu, click **Reports > Details View**.
2. Select the report in the **Reports** tree that has the associated subreport that you want to delete.
3. Click the **Properties** tab.
4. Click  in the **Actions** column of the subreport that you want to delete.
5. Click **Yes** on the confirmation dialog box to confirm the deletion.

Report Templates

Report templates define how the data of a report is rendered. For detailed information about managing your report templates, see [Managing Report Templates](#).

Because shared global reports are available in all projects for all users, user roles that are able to update and delete them require special permissions: **Manage shared global reports** and **Delete shared global reports**. These permissions also apply to report templates that are used in shared global reports. Non-shared global reports and report templates can be updated and deleted by user roles with the standard report permissions for updating and deleting reports.

Selecting or Uploading Report Templates

To select a report template or upload a template from your local system:

1. In the menu, click **Reports > Details View**.
2. Select the report to which you want to associate the template.
3. Click the **Report** tab.
4. To upload a template, click the **Click here to upload a new report template** link to open the **Upload Template** dialog box.
 - a) Give the template a meaningful **Name** and **Description**.
 - b) In the **Projects** list box, select the project to which you would like to make the template available or select **All Projects** to have the template associated with all projects.
 - c) Click **Browse** to browse to and select the template on your local system.
 - d) Click **OK** to upload the template.
5. To select an existing template, click the **Click here to choose a report template** link to open the **Select Report Template** dialog box.
 - a) Select a template from the list that is designed to properly render the data of the selected report.
 - b) Click **OK** to use the template.

Downloading Report Templates

Silk Central report templates render report data into formats that meet your specific needs. Templates can take the form of Word documents, Excel spreadsheets, BIRT Report Designer templates, XML, or CSV files. Downloading Silk Central report templates to your local system enables you to edit them through BIRT Report Designer or Microsoft Excel or Word. After you download and edit a report, you can upload it to make it available to other users. For details see the related [Uploading Report Templates](#) procedure.

To download an existing report template for editing:

1. In the menu, click **Reports > Details View**.
2. Select a report that utilizes the template you want to modify from the **Reports** tree.
3. Click the **Properties** tab.
4. Click the download link of the template you want to download.

The available download links are:

| | |
|---|--|
| Generate empty BIRT report template | You receive the report data as an empty generic BIRT report template. The data source is already configured. |
| Generate .XLS/.XLSX report template | You receive an Excel file with a sheet that contains the data. You can specify additional information in adjoining sheets, for example diagrams. |
| Download sample Word report template | You receive a Word <code>docx</code> file that you can use to create a custom Word report. The Word file contains examples of how to customize the template to add the data that you need for your report. |

Download report data as .CSV file

You receive the report data as a Comma Separated Values (CSV) file. Depending on your local settings, you will receive ',' or ';' as the delimiter character. The date is also formatted based on user settings.

Download report data as .XML file

You receive the report data as XML. The advantage of this approach over CSV is that you retain all subreport data. Accessing data outside of Silk Central - You can call a specific URL that offers the report data using the following format:

```
http://server/servicesExchange?  
hid=reportData&userName=<username>  
&passWord=<password>&reportFilterID=<ID of the  
report>&type=<csv|xml>  
&projectID=<ID of the project>
```

5. The **File Download** dialog box displays. Click **Save** and download the report file to your local system.
6. Edit the report based on your needs using either the BIRT Report Designer for `rptdesign` files, Word for `docx` files, or Excel for `xls/xlsx/xlsm` files.

Removing Report Templates

To remove the template of the current report:

1. In the menu, click **Reports > Details View**.
2. In the **Reports** tree, select the report from which you want to delete a template.
3. Click the **Report** tab.
4. Click .
5. Click **Yes** on the subsequent confirmation dialog box.

Report Subscriptions

The **Subscriptions** tab in the reporting unit allows you to subscribe to reports. You can customize when, where, and how you receive the report as well as modify report parameters. The report is created in the background on the application server, per a selected schedule, and the subscriber is sent the report via email or the report can be displayed in a dashboard panel and is also accessible from external sites through a URL.

Managing Report Subscriptions

Please note the following when working with subscriptions:

- If the user who created the report is deleted, the schedule for the report is removed, disabling the report. Another user must modify the subscription to re-activate it.
 - Subscriptions are not copied when you copy or baseline a project.
 - Subscriptions are not exported when you export a project.
 - After a subscription is created, you can edit it or delete it using icons in the **Actions** column.
 - For dashboard panel and externally accessed reports, only the last report result is stored and there is no history available.
1. In the menu, click **Reports > Details View**.
 2. Select a report in the **Reports** tree.
 3. Click the **Subscriptions** tab.
 4. Click **Add Subscription**. The **Edit Subscription** dialog box opens.
 5. Give your subscription a name in the **Name** field.

6. To modify the subscription parameters, click **Edit Parameters**. The **Edit Parameter** dialog box opens. Change the **Value** and click **OK**.
7. Select a schedule from the **Schedule** list. If the schedule you want does not exist, you need to create it. You can also select no schedule, for example for subscriptions that will only be triggered based on the outcome of a test execution.
8. In the **Result Handling** area, pick one of the following options:
 - If you want to send the report as an email, click **Send as Email** and select the **Format**: Report template (Excel, BIRT), PDF (data table) or XLSX (data table). The available options depend on the existence of the configured template. Then, enter the **Email addresses** where the report will be sent. You can send the report to multiple email addresses by separating them with commas, semi-colons or blanks.
 - If you want to make the report available in a dashboard panel or from external sites, click **Provide for dashboard and external access**. Selecting this option makes the report available in the **Report Subscription** dashboard panel, and you can embed the provided link in Websites, emails, and more. This is a permanent link that does not change and requires no login or license to access. Note that when using a template, only BIRT templates are graphically rendered. Other templates display as data table.
9. Click **OK**. The new subscription is listed in the **Subscriptions** list.

To manage all report subscriptions, select the **Global Reports** node or the root node of the project. Use the **Actions** column icons to edit or delete report subscriptions. Because shared global reports are available in all projects for all users, user roles that are able to update and delete them require special permissions: **Manage shared global reports** and **Delete shared global reports**. These permissions also apply to report templates that are used in shared global reports. Non-shared global reports and report templates can be updated and deleted by user roles with the standard report permissions for updating and deleting reports.

Adding a Report to a Dashboard Panel

Using any report available in the **Reports** unit, the **Report Subscription** panel of your dashboard allows you to display an existing report. First, create and configure your report, and then add the panel to your dashboard.

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Subscriptions** tab.
4. Click **Add Subscription**. The **Edit Subscription** dialog box opens.
5. Give your subscription a name in the **Name** field.
6. To modify the subscription parameters, click **Edit Parameters**. The **Edit Parameter** dialog box opens. Change the **Value** and click **OK**.
7. Select a schedule from the **Schedule** list. If the schedule you want does not exist, you need to create it. You can also select no schedule, for example for subscriptions that will only be triggered based on the outcome of a test execution.
8. Click **Provide for dashboard and external access** to make the report available in a dashboard panel.
9. Click **OK**. The new subscription is listed in the **Subscriptions** list.
10. In the menu, click **Home > My Dashboard**.
11. Click **Add Panel** on the top left. The **Add Panel** dialog box appears, listing the available panels along with a short description for each panel.
12. Select **Report Subscription**. The **Configure Panel** dialog box appears, listing all available report subscriptions.
13. Select a report subscription from the list and click **OK**. The panel is added to your **Dashboard**.

Working with Charts

Displaying Charts

To display a chart:

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Chart** tab to display the default chart.
4. To select a different chart type, click . The **Select Chart Type** dialog appears.
5. Select a chart type from the **Chart type** list.
6. Check the view properties that you want to apply to the chart:
 - 3D view
 - Show horizontal grid lines
 - Show vertical grid lines
 - Show legend
7. Specify how these chart options are to be saved:
 - Click the **For current user only** option to have these chart settings override the report's standard settings whenever the current user views this chart.
 - Click the **As report standard** option to have these chart settings presented to all users who do not have overriding user settings defined. This setting does not effect individual user settings.
8. Click **OK** to display the new chart type.



Note: The chart configurations you define here become the default for this report. When standard charts and graphs are not able to deliver the specific data that you require, or when they cannot display data in a required format, you can customize the appearance of queried data using the Silk Central reporting functionality. To open the current chart in a separate browser window, click  at the top of the **Chart** page.

Printing Charts

To print the current chart:

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Chart** tab.
4. Click **Print**. The chart data displays in a new window in printable format. Your system's print dialog box is also displayed.
5. Configure print settings as necessary and click **OK** to print the chart.

Removing Charts

Removing a chart only removes the currently selected chart template from the selected report, it does not remove the chart template entirely.

To remove the current chart template from the selected report:

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Chart** tab.
4. Click **Remove chart type**. The **Remove Chart** dialog box opens.
5. Select one of the following:

- Select **Remove user settings (and revert to report standard)** to have the current user's chart settings deleted along with the chart. The chart will subsequently be displayed according to the report's standard settings. If no standard settings have been defined, the chart cannot be displayed.



Note: This option is only available when the current user has defined specific chart settings.

- Select **Remove standard chart settings** of report to have any standard settings deleted along with the chart. User-specific settings are not affected by this option.



Note: This option is only available when standard chart settings have been defined for a report.

6. Click **OK** to delete the chart template. If required, you can click the **<Click here to choose a chart type>** link to assign a new chart template to the selected report.

Report Chart Page

Reports > Details View > Chart

The **Chart** page enables you to define charts and graphs for data analysis. The page relies on the internal reporting engine of Silk Central to create standard charts and graphs from the data retrieved by the selected report query.

The following standard chart types are available:

- area chart
- bar chart
- horizontal stacked bar chart
- line chart
- pie chart
- stacked bar chart

Customizing Reports

Customizing BIRT Report Templates

With BIRT Report Designer (BIRT), you can customize the pre-installed report templates of Silk Central and create custom report templates. For details on using BIRT, see the *Administration* topics in this Help and the BIRT Report Designer documentation.

To download an existing template for editing:

1. In the menu, click **Reports > Details View**.
2. Select a report that utilizes the BIRT Report Template.
3. Click the **Properties** tab.
4. Click **Generate empty BIRT report template**. You receive the report data as an empty generic BIRT report template. The data source is already configured.
5. Once you have saved the template to your local system, modify it as required.

For detailed information on configuring BIRT report templates, see the *Administration* topics in this Help.

6. To upload the modified report template, click **Administration > Report Templates** in the menu and click **Upload**.

Customizing the Word Report Template

Use the Word report template to create a custom report that suits your needs.

To download an existing template for editing:

1. In the menu, click **Reports > Details View**.

2. Select a report.
3. Click the **Properties** tab.
4. Click the **Download sample Word report template** link. Use your browser to save the file locally.
5. Using the available result columns in your report, customize the report to your liking according to the commands contained in *Word Report Template*.
6. To upload the modified report template, click **Administration > Report Templates** in the menu and click **Upload**.

Word Report Template

About the template

Use the Word report template to create a customized report. The template uses Word merge fields to get the data that you need. The template is designed for `docx` files and does not support `doc`. Here a few tips to get started:

- To insert merge fields in Microsoft Word, click **Insert > Quick Parts > Field > MergeField** or click **Ctrl +F9**.
- In order to edit a merge field, right click the field and then select **Edit Field**.
- You can mix and match standard text in addition to merge fields in your report.
- Merge fields take the following form: `<< value >>`.

General Report Information

The following contains data you may want to use in your report header:

| Field | Description |
|---|---|
| <code>\${report.name}</code> | The name of the report. |
| <code>\${report.projectName}</code> | The name of the project where the report is executed. |
| <code>\${report.description}</code> | The description of the report. |
| <code>\${report.executedBy}</code> | The login name of the user who executed the report. |
| <code>\${report.generatedOn}</code> | The date and time when the report was executed. |
| <code>\${report.generatedOnWeek}</code> | The week number of when the report was executed. |

Accessing Data

The following sections describe how to access and display different data for your report.

By Index `<<${rows[0].cols[3]}>>`

By Name `<<${rows[0].ReqName}>>`

In Lists A typical use case would be to iterate through each row and show the column information for every row. To get a list of data, do the following:

```
<[#list rows as r]>
<${r.TestDefID}> - <${r.TestName}>
</#list>
```

The beginning of the loop is defined by `[#list rows as r]` and the end by `[/#list]`. The row data within the loop is referenced with `r` which allows you to get `Test Name` within the loop by referencing it with `${r.TestName}`.

The above example would iterate through all rows and the rendered report would show a list of tests with their id and their name.

In Tables Defining the start and end of a loop within a table row requires an additional command to continue iterating within the table. You have to place the `list` command for the loop into the first column of your table and preface it with `@before-row`. You need to use `@after-row` to complete the list. The following table shows how to do this.

| | |
|--|---|
| <code><<{{headers[3]}} <{{headers[3].@name}}>></code> | <code><<{{headers[4]}} <{{headers[4].@index}}>></code> |
| <code><<@before-row[#list rows as r]>><{{r.ReqName}}>></code> | <code><{{r.ReqDescription.@html}}>></code> |
| <code><<@after-row[#list]>></code> | |

Headers and HTML

- Headers of the data can be accessed with a merge field named `headers`.
- Headers can be displayed in different ways by appending `.@elementName` or `.@name` or `.@index`.
- Some Silk Central data, like many **Description** fields, are stored as HTML. If you want to display your HTML-encoded text based on its tags, append the attribute `.@html`. If you don't add this attribute, you will still see the text, but it will contain the HTML markup. For example: if your source data is `My report description` and you use `.@html`, you will see `My report description`. Otherwise, you will see: `My report description`. The following HTML tags are supported: ``, ``, `<i>`, `<u>`, `
`, `<p>`, and `<a>`.

Subreports

Subreports can be accessed with the keyword `sub`. Append the number of the subreport that you want to access, for example `sub1`, `sub2`, etc.

Example: If `ProjectID` is in the first column of your main report, refer to it with `<${rows[0].ProjectID}>`. If `TestID` is in the first column of the first subreport, refer to it with `<${sub1.rows[0].TestID}>`. To iterate through all rows of the first subreport and show the `TestID` column and the second column:

```
<[#list sub1.rows as sr]>
  <${sr.TestID}> <${sr.cols[1]}>
</#list>
```

Troubleshooting

Why does my report show wrong data?

Are you referencing columns by their index? If yes, be aware that the first column is referenced by index 0 and not by 1.

Why do I get an error mentioning a specific field when downloading my report? I don't see this field in my template?

You may have changed the label of a field but the mail merge reference still has its original value. For example, you may receive an error message that the field `${r.requirementName}` doesn't exist. In your document you see the field `${r.reqName}`. When you right click this field and select **Edit Field**, you will see that the reference is still `${r.requirementName}`. Don't forget to change mail merge fields via **Edit field**.

Sample Report

The following sample Word report shows a report with manual tests containing steps.



Note: You cannot copy and paste this sample as is. It is designed to show you different ways to use reporting code for the various sections in a report. The sample shown below is included in the **Report Templates** as `ManualTestResults.docx`.

```
Report generated: <${report.generatedOn}> <${report.generatedOnWeek}>
<${report.name}>

Project Name          <${report.projectName}>
Report Description    <${report.description}>
```


1. In the menu, click **Reports > Details View**.
2. In the **Reports** tree, select the report that you want to save.
3. Click the **Report** tab.
4. Click  on the **Report view** toolbar.
5. On the **File Download** dialog box, click **Save** to save the PDF document to a location of your choice.

Bookmarking Reports

Select **<user name> > Copy Bookmark URL** in the Silk Central menu to bookmark the currently displayed report, including the parameters that you have set in the **Parameters** page. You can send bookmark URLs to other Silk Central users, allowing them to view reports with a single click.

The bookmark URL contains the parameters, prefixed with `rp_`. Date values are represented as the correlating `Long` values in UTC in the URL.

Viewing a Report as a PDF

To view the current report in PDF format within the report browser frame:

1. In the menu, click **Reports > Details View**.
2. In the **Reports** tree, select the report that you want to view.
3. Click the **Report** tab.
4. Click  on the report view toolbar. The report displays in PDF format.

Accessing Most Recently Used (MRU) Reports

To select a recently-viewed report:

1. In the menu, click **Reports > Details View**.
2. On the **Reports** toolbar, expand the **Last Used Reports** list box.
3. Select the report that you want to view.

Reports Toolbar Functions

Silk Central > Reports

The **Reports** toolbar provides important commands for report management. The toolbar includes the following items:

| Item | Icon | Description |
|-------------------------|---|--|
| New Child Folder |  | Enables creation of new report folders. Click New Child Folder to define a name and optional description for a new folder. The new folder displays as a child of the currently selected node in the Reports tree. |
| New Child Report |  | Enables creation of new reports. Click New Child Report to define a new report using the Create New Report dialog box. A new report displays as a child of the currently selected node in the Reports tree. |
| Edit |  | Edit a report. |
| Delete |  | Delete a report. |
| Cut |  | Cut a report and add it to the clipboard. |
| Copy |  | Copy a report to the clipboard. |

| Item | Icon | Description |
|-------------------|---|--|
| Paste |  | Paste a report from the clipboard into the Reports tree. |
| Paste as Child |  | Paste a report from the clipboard into the Reports tree, as a subnode of the selected node. |
| Move Up |  | Move reports up within the Reports tree. |
| Move Down |  | Move reports down within the Reports tree. |
| Last Used Reports | | Lists the Most Recently Used (MRU) reports by date/time in descending order. Select a report name from the list to advance to that report. Each time a report is accessed by clicking the Data , Chart , or Report tab, that report is added to the top of the list box. Accessing a report's Properties or Parameters tab does not result in that report being added to the Last Used Reports list box. The Last Used Reports list box is empty for new users and users who have not yet generated a report. The number of reports that displays in this list can be configured by your administrator. For more information, see the <i>Administration</i> topics in this Help. |

Report Properties

Editing Report Properties

To edit the properties of a report:

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Properties** tab.
4. Click **Edit**. The **Edit Report** dialog box appears.
5. Modify the **Name**, the **Description** and the **Timeout [s]** of the report as required.
6. Check the **Share this report with other users** check box if you want to make this report available to other users.
7. From the **Default tab** list, select the tab that you want to be directed to when you select this report from one of the context-sensitive report lists.
8. You can edit the report in two ways:
 - Create a simple report: Use the **Selection criteria**, **Property**, **Operator**, and **Value** lists to generate SQL queries. Click **More** to add further query strings and choose the operators **AND** or **OR** to combine the queries. Click  to delete a query string.
 - Create an advanced report: If you are familiar with SQL, you may want to edit the query code. Click **Advanced Query** and modify the query code within the **Report data query** field. The **Insert placeholder** list assists you in editing the SQL queries with predefined function placeholders. Click **Simple** to go back to the simple mode.



Note: If you manually edit the SQL code for the query, upon finishing, click **Check SQL** to confirm your work.

9. Click **Finish** to save your changes.

Report Properties Page

Reports > Details View > Properties

The **Properties** page displays the basic properties of the selected report, enabling you to edit these properties or the report templates. You can also add subreports to your reports.

The page displays the following items:

| Items | Description |
|----------------------|---|
| Report Name | Name of the report (customizable) |
| Report ID | System-defined identifier of the report |
| Description | A description of the report (customizable) |
| Created On | Date the report was created. Default reports are created when a database is created and connected to. |
| Created By | User who created the report. Default reports are created by the user Admin. |
| Changed On | Date the report was last modified. |
| Changed By | User who last modified the report. |
| Renderer | Report template that is currently assigned to the report. |
| Default Tab | Tab you are directed to when you select this report from one of the context-sensitive report lists. |
| Edit | Click to open the Edit Report dialog box. |
| Add Subreport | Click to add a subreport to the report. |
| Report Templates | <p>The available pre-installed report templates are:</p> <p>Generate empty BIRT report template You receive the report data as an empty generic BIRT report template. The data source is already configured.</p> <p>Generate .XLS/.XLSX report template You receive an Excel file with a sheet that contains the data. You can specify additional information in adjoining sheets, for example diagrams.</p> <p>Download sample Word report template You receive a Word docx file that you can use to create a custom Word report. The Word file contains examples of how to customize the template to add the data that you need for your report.</p> <p>Download report data as .CSV file You receive the report data as a Comma Separated Values (CSV) file. Depending on your local settings, you will receive ',' or ';' as the delimiter character. The date is also formatted based on user settings.</p> <p>Download report data as .XML file You receive the report data as XML. The advantage of this approach over CSV is that you retain all subreport data. Accessing data outside of Silk Central - You can call a specific URL that offers the report data using the following format:</p> <pre>http://server/servicesExchange? hid=reportData&userName=<username> &passWord=<password>&reportFilterID=<ID of the report>&type=<csv xml> &projectID=<ID of the project></pre> |

Report Parameters

Editing Report Parameters

To edit the parameters of a report:

1. In the menu, click **Reports > Details View**.
2. Select a report in the **Reports** tree.
3. Click the **Parameters** tab. If the report has parameters defined for it, the parameters are listed here.
4. Click **Edit Parameters**. The **Edit Parameters** dialog box appears.
5. Edit the **Label** or **Value** of the listed parameters as required.
6. Click **OK**.

Report Parameters Page

Reports > Details View > Parameters

The **Parameters** page lists customizable statement elements. Parameters can be defined any time before a report execution by simply changing them on the **Parameters** page. The syntax of a parameter is: `{parametername|defaultvalue|guiname}`. The `defaultvalue` and the `guiname` are optional. Parameter-names cannot contain whitespace characters.

When a report has parameters associated with it, it is possible to edit the values of the parameters before each report execution. Parameter values are stored in the current user context, which means edited values are available only to the user who performs the edits. When parameter values are not specified for a given report execution, the default values from the report definition are used.

When a report has subreports assigned to it, the parameters of those subreports are also shown in the **Parameters** page and the values are stored only within the context of the selected report. For example, the values are only used in conjunction with the current subreport configuration. When creating new reports, parameters are the values that are defined on the **Create New Report** dialog box in the **Selection criteria** area.

Report Data Page

Reports > Details View > Data

The **Data** page serves as a read-only result preview that shows the results of the selected report a dynamic grid with sorting and filtering capabilities. The visible data can be exported to Excel with the **Download Data (.xlsx)** button.

You can access the requirements, tests, execution plans, and Issue Manager issues that you query for directly from the results list in the **Data** page. To do so, your query must include the column `ProjectID` and the respective ID of the element that you want to link to.

| ID | Description |
|------------------|---|
| RequID | Query for this column to enable a link to requirements on the Data page of a report. |
| TestDefID | Query for this column to enable a link to tests on the Data page of a report. |
| ExecDefID | Query for this column to enable a link to execution plans on the Data page of a report. |
| DefectID | Query for this column to enable a link to Issue Manager issues on the Data page of a report. |

If the query's result includes both the `ProjectID` and either of `RequID`, `TestDefID`, `ExecDefID`, or `DefectID` using exactly these terms as column names, the **Data** page will display the values in the element ID's column as a link.

Report Page

Reports > Details View > Report

The **Report** page is used to display data as a formatted report. If you have not yet assigned a template for your report, you can select one in the **Report** page.

A list box provides a selection of all available report templates. In addition to many system-installed templates, any custom report templates that were uploaded from **Administration > Report Templates** are also available here.

You can also download an existing template by clicking the **Properties** tab, and then clicking the download link that corresponds to the report format you are working with. From there you can customize the template to your needs.



Note: Reports are cached to improve the performance of reporting. Click **Update** to update the report data immediately.

Project Overview Report

Reports > Project Overview Report

Displays the **Project Overview Report**, which offers a high-level overview of the status of the selected project. The **Project Overview Report** includes the following sections:

| Section | Description |
|----------------------------|---|
| General Report Information | General information like the name of the current project, the report description, and the planned release date. |
| Requirements | Test coverage status for requirements. Shows the percentage of test coverage in tabular and graph format both all requirements and requirements with high priority. |
| Tests | Test-type distribution and test-execution results in chart and tabular form for tests. |

Code Analysis

The goal of conventional code coverage is to deliver information about what code is covered by tests. This approach is typically used to gain code-coverage information for unit tests. The code-analysis functionality in Silk Central goes well beyond this, delivering data for unit, functional, and load tests, both automated and manual, in managed environments. Code coverage measurements are utilized to track test progress and guide test planning. This innovative approach in Silk Central to code coverage draws on the relationship between specific tests and the code they test. This approach enables you to perform impact/dependency analysis of code changes from the testing perspective. It also assists you in optimizing your testing, by helping you identify the test runs that are most relevant to a specific code change.

The **Code Analysis** unit offers code-coverage data for AUT (Applications Under Test), packages/namespaces, classes, methods, and statements, enabling you to perform test-impact analysis, which determines the tests that should be run in response to specific code changes, and effort analysis, which determines how many hours of automated testing and manual testing are required to adequately cover specific code changes. Each of these code-analysis tasks can be addressed by running pre-configured reports. The unit features a navigation tree that lists all products that have been created for the selected project. You can drill down into products to select specific versions, and at the deepest level, specific builds.

Enabling Code Analysis

You must configure Silk Central to gather code coverage data from an application under test. You can configure any number of execution plans listed in **Executions > Details View > Deployment**.



Tip: Make sure that only code coverage-enabled tests access your application under test and that no parallel tests are executed. This ensures that the code coverage data is correct and that it can be assigned to a specific test later.

Java Code Analysis Options

To use Java Code Analysis for an AUT, you have to run the JVM of the AUT with a Java Agent. You can download the Java Agent package by selecting **Java Code Analysis Agent** under **Help > Tools** in the Silk

Central menu. Java Code Analysis requires the latest version of the Java Runtime Environment (JRE) 1.8 to be installed. You can use the following options for all supported Java versions:

| Option | Description |
|---|--|
| <code>port=19129</code> | <i>Optional:</i> Port of code coverage service. When no port is specified the port 19129 is used. |
| <code>coveragepath=[path1], [path2], ...</code> | Pathes to the .jar files that are covered by the agent. If parts of a path contain whitespaces, you have to use quotes. For example: <code>coveragepath="C:\Program Files\YourApplication\"</code> . If a folder is specified, the agent recursively looks for all .class files in this folder. If a folder with a trailing wildcard is specified, the agent looks for .jar files in this folder. Running the agent without specifying a coveragepath is not valid. <pre>java -javaagent:agent.jar=coveragepath="C:Program Files\YourApplication\" Aut.jar</pre> |
| <code>includes=[package1]: [package2]: [package3]: ...</code> | <i>Optional:</i> Only packages that match these packages are considered for coverage. Wildcards (*) are allowed for specifying the included packages. <pre>java -javaagent:agent.jar=includes=com.borland.* AUT.jar</pre> |
| <code>excludes=[package1]: [package2]: [package3]: ...</code> | <i>Optional:</i> Packages that are specified in the excludes option are ignored for code coverage. Wildcards (*) are allowed for specifying the excluded packages. <pre>java -javaagent:agent.jar=includes=com.borland.*; excludes=com.borland.internal* AUT.jar</pre> |



Note: When both inclusions and exclusions are specified in the filters, the exclusions are evaluated first. If a class is both included and excluded through filters, this class will be excluded and no coverage data is generated for this class. If the filter option is not specified, coverage data is generated for all used classes, unless the coverage path option is set, in which case coverage data is generated for all classes in the coverage path.

Java Code Analysis Example

Example for Windows:

```
"<java_home_directory>\bin\java" -
javaagent:agent.jar=port=19129;
coveragepath=C:\dev\deploy\lib\library1.jar,C:\dev\deploy\lib
\library2.jar;
includes=com.borland.*;excludes=com.borland.internal.*
com.borland.MyApplication
```

Example for Linux (including quotation marks):

```
"<java_home_directory>/bin/java" -
javaagent:agent.jar="port=19129;
coveragepath=/tmp/dev/deploy/lib/library1.jar,/tmp/dev/
deploy/lib/library2.jar;
includes=com.borland.*;excludes=com.borland.internal.*"
com.borland.MyApplication
```

.NET Code Analysis

Installing the Windows Code Analysis Framework and DevPartner Analytics

Before you can perform this task, make sure that Microsoft .NET Framework 3.5 is installed on the machine where the Code Analysis Framework will be installed (the machine that hosts the application to test). Older or newer versions of Microsoft .NET Framework will not work!

1. Go to the machine that hosts the application to test.
2. Open Silk Central.
3. Go to **Help > Tools** and click **Windows Code Analysis Framework**. This will prompt a download of `CAFramework.exe`. `CAFramework.exe` includes the installation for the Code Analysis Framework and the DevPartner Analytics package.

 **Note:** Multiple execution servers running tests against the same Code Analysis Framework at the same time is not supported.

4. Double click the download file to start the installation. Walk-through the installation of the **Windows Code Analysis Framework**. At the end of the installation, you will be prompted to install `DevPartner Analytics`. Ensure that the **DevPartner Analytics Installation** check box is checked on the appropriate installation screen and install this component. At the end of this installation, all required files are installed and the proper service is started. The name of the service is `Silk Central Code Analysis Service`.

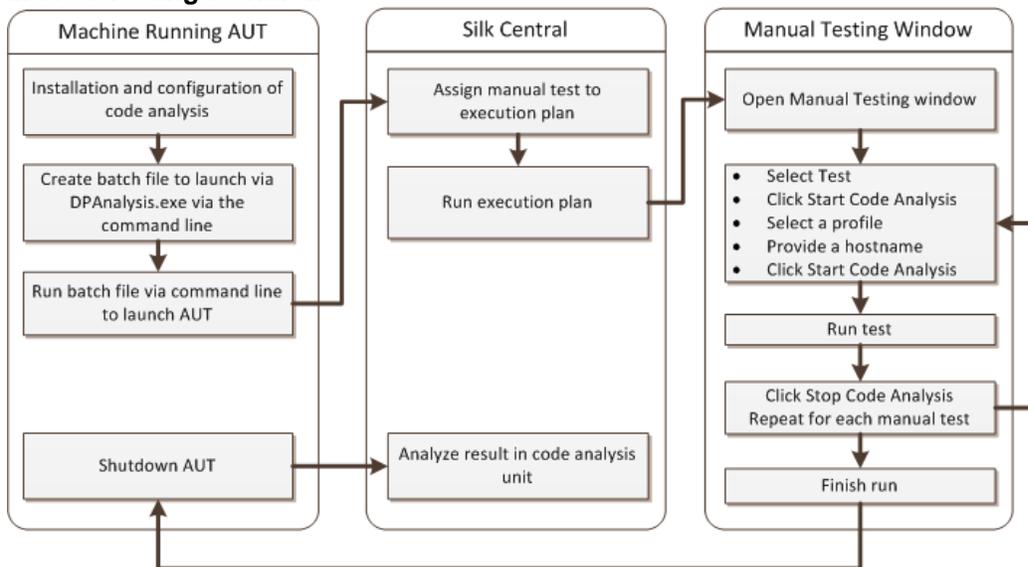
.NET Code Analysis and Manual Testing

The following sections detail the required steps to install and use the DevPartner .NET code coverage components shipped with Silk Central. The example below will get you started by using manual testing and a single AUT. If you plan to use automated testing, first read and understand the steps below, and then see the topic titled *.NET Code Analysis and Automated Testing*.

 **Note:** In order to use the Silk Central/.NET code coverage functionality, the Micro Focus DevPartner Analytics components version 11.2 or later must be used.

 **Note:** .NET code analysis requires the `PDB` files generated at compile time of the .NET application under test (AUT) to be located in the same directory as the AUT's executable(s). If the `PDB`s for the AUT are not available, it is not possible to leverage the .NET code analysis functionality.

Manual Testing Workflow



Configuring DPAnalysis to launch Your AUT

1. Create a .BAT file that points to DPAnalysis.exe. An example would be:

```
"C:\Program Files\Micro Focus\DevPartner\Analysis\dpanalysis.exe" /cov /  
USE_AUTOMATION_INTERFACE  
/W "C:\Common\Development\MYAPP"  
/P "C:\Common\Development\MYAPP\MyApp.exe"
```

2. For readability, the example above is on multiple lines. Your .BAT file should be on one line. Make the following modifications:
 - a) Ensure that the first line is your installation path to dpanalysis.exe. The two switches are mandatory.
 - b) The second line contains a mandatory switch and needs to include the path to your executable and PDBs.
 - c) The third line contains a mandatory switch and needs to include the absolute path to your test application and the application name.
3. Double-click the .BAT file to have DPAnalysis launch your application.

For more information on running DPAnalysis.exe, see the topic titled *Using DpAnalysis.exe*. That topic and related topics contain information on additional switches and how to use a configuration file if you have multiple applications to test.

Configuring Silk Central for Code Analysis

1. In the menu, click **Execution Planning > Details View**.
2. Open an existing execution plan that contains a manual test with steps to test functionality for your AUT.
3. Click the **Deployment** tab.
4. Under **Code Analysis Settings**, click **Edit** to view the **Edit Code Analysis Settings** dialog box.
5. Ensure that **Enable Code Analysis** is checked.
6. Select DevPartner .NET Code Analysis from the **Code Analysis Profile** list.
7. For the **Hostnames** field, type in the host name followed by colon and the port number. For example: localhost:19129.

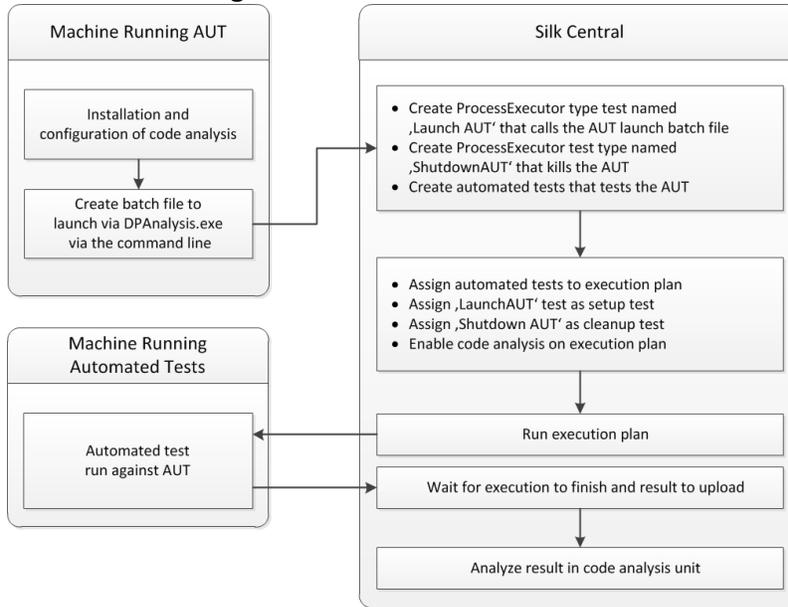
The default code analysis framework port is 19129. You can configure the code analysis framework to run at a given port using the following xml file: <install dir>\Code Analysis \CodeAnalysisConfig.xml
8. Click **OK**.
9. Run the test by clicking **Run** in the **Execution** area.
10. On the **Run** dialog box, ensure that **All Tests** is checked.
11. In the **Set build for execution plan**, select the proper build for your execution plan.
12. Click **OK**.

.NET Code Analysis and Automated Testing

This section describes two scenarios that you may encounter when executing your tests that use code coverage. It also provides proper ways to structure your tests within an execution plan. The examples used in the following topics are specific to DevPartner Code Analysis, but the structure is similar for other code analysis tools. For example, you will have different startup/cleanup parameters in your execution plan.

 **Note:** There may be a delay between starting the AUT and starting code analysis, which may lead to some of the start-up code of the AUT not being recorded by code analysis.

Automated Testing Workflow



Configuring One or More Tests with One AUT

If you have the scenario where you have one or more tests in a single execution plan and they test against a single application under test, you should structure your testing as follows:

1. Create a `ProcessExecutor` test that will start your code analysis application.

The `ProcessExecutor` **Argument List** should start your code analysis program.

For example:

- Executable Name:

```
cmd
```

- Argument List:

```
/c start cmd
/c D:\sctm\dps\startDPS_Simple1.bat
```

Where `startDPS_Simple1.bat` contains:

```
"C:\Program Files (x86)\Micro Focus\DevPartner\Analysis\dpanalysis.exe" /
cov /USE_AUTOMATION_INTERFACE /O "D:\SCTM\temp" /W "D:\SCTM\DPS\TESTAPP
\SCTMCodeAnalysisTestApp\bin\Debug" /P "D:\SCTM\DPS\TESTAPP
\SCTMCodeAnalysisTestApp\bin\Debug\SCTMCodeAnalysisTestApp.exe"
```

2. Create an execution plan.
3. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Setup Test** row.
4. Add the previously created test as the **Setup Test**.
5. Add a test to the execution plan that automates your application under test with an automated testing tool.
6. *Optional:* Add another test to the execution plan that automates your application under test with an automated testing tool.
7. Create a `ProcessExecutor` test.

The `ProcessExecutor` **Argument List** should close your program:

- Executable Name:

```
cmd
```

- Argument List:

```
/c TASKKILL /IM SCTMCodeAnalysisTestApp.exe /F
```

8. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Cleanup Test** row.
9. Add the previously created test as the **Cleanup Test**.

Configuring One or More Tests with Multiple AUTs

If you have the scenario where you have one or more tests in a single execution plan and they test against multiple AUTs, you should structure your testing as follows:

1. Create a `ProcessExecutor` test that will start your code analysis application.

The `ProcessExecutor` **Argument List** should start your code analysis program. For example:

- Executable Name:

```
cmd
```

- Argument List:

```
/c start cmd
/c D:\sctm\dps\two_app_dps.cmd
```

Where `two_app_dps.cmd` contains:

```
"C:\Program Files (x86)\Micro Focus\DevPartner\Analysis\dpanalysis.exe" /
config D:\SCTM\DPS\configuration_file.xml
```

and `configuration_file.xml` contains:

```
<?xml version="1.0" ?>
<ProductConfiguration>
  <RuntimeAnalysis Type="Coverage" MaximumSessionDuration="1000"
NoUIMsg="true" />
  <Targets RunInParallel="true">
    <Process CollectData="true" Spawn="true" NoWaitForCompletion="true"
NM_USE_AUTOMATION_INTERFACE="1" >
      <AnalysisOptions NM_USE_AUTOMATION_INTERFACE="1" NO_MACH5="1"
NM_METHOD_GRANULARITY="1" SESSION_DIR="c:\temp" />
      <Path>D:\SCTM\DPS\SCTMCodeAnalysisTestApp
\SCTMCodeAnalysisTestApp\
SCTMCodeAnalysisTestApp\bin\Debug\SCTMCodeAnalysisTestApp.exe</
Path>
      <Arguments></Arguments>
      <WorkingDirectory>D:\SCTM\DPS\SCTMCodeAnalysisTestApp
\SCTMCodeAnalysisTestApp\
SCTMCodeAnalysisTestApp\bin\Debug</WorkingDirectory>
    </Process>
    <Process CollectData="true" Spawn="true" NoWaitForCompletion="true"
>
      <AnalysisOptions NM_USE_AUTOMATION_INTERFACE="1" NO_MACH5="1"
NM_METHOD_GRANULARITY="1" SESSION_DIR="c:\temp" />
      <Path>D:\SourceCode\SVNQAD\Development\Development\SOURCECODE\
QADSCTMMigration\bin\QADSCTMMigration.exe</Path>
      <Arguments></Arguments>
      <WorkingDirectory>D:\SourceCode\SVNQAD\Development\Development\
SOURCECODE\QADSCTMMigration\bin</WorkingDirectory>
    </Process>
  </Targets>
</ProductConfiguration>
```

2. Create an execution plan.
3. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Setup Test** row.

4. Add the previously created test as the **Setup Test**.
5. Add a test to the execution plan that automates your first application under test with an automated testing tool.
6. Add another test to the execution plan that automates your second application under test with an automated testing tool.
7. Create a `ProcessExecutor` test.

The `ProcessExecutor` **Argument List** should close your programs:

- Executable Name:

```
cmd
```

- Argument List:

```
/c TASKKILL /IM application1.exe /IM Application2.exe /F
```

8. Click **Execution Planning > Details View**, select an execution plan, expand the **Test Execution Setup** area, and click **Edit** in the **Cleanup Test** row.
9. Add the previously created test as the **Cleanup Test**.

Using `DPAnalysis.exe`

`DPAnalysis.exe` is the DevPartner code coverage application that allows you to profile .NET applications. The application is started using the command line.

You can use the application with traditional switches as documented in these topics, or you can add an additional switch that points to a configuration file. You may want to use a configuration file if you have an exceptionally long command line string or you want to run multiple applications in a single session.

The `/USE_AUTOMATION_INTERFACE` switch is mandatory. It causes the automation interface environment needed for the interaction with the framework to be established. It also causes license checking to be bypassed. This switch does not take any parameters.

Running `DPAnalysis.exe` from the Command Line

The DevPartner Studio Code Coverage installation includes `DPAnalysis.exe`, a command line executable that is installed in the `\Program Files (x86)\Micro Focus\DevPartner\Analysis\` directory.

Use the following syntax and switches to run the DevPartner Studio Code Coverage tool from the command line:

```
DPAnalysis [/Cov] [/USE_AUTOMATION_INTERFACE] [/E|/D|/R]
[/W workingdirectory] [/PROJ_DIR] [/H hostmachine]
[/NOWAIT] [/NO_UI_MSG] [/N] [/F] [/A C:\temp1;C:\temp2]
[/NO_QUANTUM /NM_METHOD_GRANULARITY /EXCLUDE_SYSTEM_DLLS
/NM_ALLOW_INLINING /NO_OLEHOOKS /NM_TRACK_SYSTEM_OBJECTS]
{/P|/S} target [target arguments]
```

For detailed information on the available switches and about how to use a `DPAnalysis` configuration file, refer to the DevPartner documentation.



Note: `DPAnalysis.exe` does not instrument unmanaged code. To collect performance or coverage analysis data for an unmanaged application, you must first instrument the application.

Example

You can use `DPAnalysis.exe` directly from the command line, using switches to direct the analysis session. For example, the following command line launches a code coverage session for the application `SCTMCodeAnalysisTestApp.exe`.

```
dpanalysis.exe /cov /USE_AUTOMATION_INTERFACE /W "D:\SCTM
\DPSCoverageIntegration\Test App\
SCTMCodeAnalysisTestApp\SCTMCodeAnalysisTestApp\bin\Debug" /P
```

```
"D:\SCTM\DPSCoverageIntegration\Test App\SCTMCodeAnalysisTestApp
\SCTMCodeAnalysisTestApp\bin
\Debug\SCTMCodeAnalysisTestApp.exe"
```

Enabling Code Analysis for Execution Plans

To enable code analysis for an execution plan:

1. In the menu, click **Execution Planning > Details View**.
2. In the **Execution** tree, select an execution plan.
3. Click the **Deployment** tab.
4. In the **Code Analysis Settings** section of the **Deployment** page, click **Edit**. The **Edit Code Analysis Settings** dialog box displays.
5. Check the **Enable code analysis** check box.
6. Select a profile from the **Code Analysis Profile** list box.
7. In the **Hostnames** field, enter a comma-separated list of host names, with port, if default port 19129 is not used, from which code analysis information is to be gathered.

For example `labmachine1, labmachine2:8000, 198.68.0.1`. For each execution plan, you need to define the host names of the machine resources where the AUT is running. For example, with a client/server system, you must not only gather code coverage information on the client, which probably runs directly on an execution server, but also on the server, which likely runs on a different machine. This applies to all multi-tiered applications.



Note: For JUnit code analysis runs, you do not need to specify a hostname.

8. Click **OK** to save your settings.



Note: Once code analysis has been defined for an execution plan, each future run of that execution plan will gather code coverage information from the defined host names. While monitoring an execution on the **Activities** page, you will see that after gathering the sources for tests, Silk Central gathers full code coverage information before beginning test runs. The Code Coverage Controller, which is integrated into each Silk Central execution server, controls all defined hosts during execution runs. For each test of an execution plan, the controller starts and stops all associated instances, collects XML-based code coverage files for the test, and merges the results into a single file. The test then saves the merged code coverage file to its execution results.

Code Analysis Details Page

Reports > Code Analysis > Details

The **Details** page displays code-coverage information for selected products, versions, and builds at the product, package, and class levels.

Product level view displays a list of covered and not-covered packages for specific products and product builds. By clicking a package name in **Product** view you can drill down to view code-coverage information for the classes that are included in that package.

Product view displays the following attributes for a selected product:

| Attribute | Description |
|---------------------|--|
| Name | Product name |
| Statements | Total amount of statements |
| Packages/Namespaces | Histogram bar view that includes the following figures: <ul style="list-style-type: none">• Total percentage of packages/namespaces that are covered |

| Attribute | Description |
|-----------|--|
| Classes | <ul style="list-style-type: none"> • Number of covered packages/namespaces, in green • Number of uncovered packages/namespaces, in red <p>Histogram bar view that includes the following figures:</p> <ul style="list-style-type: none"> • Total percentage of classes that are covered • Number of covered classes, in green • Number of uncovered classes, in red |
| Methods | <p>Histogram bar view that includes the following figures:</p> <ul style="list-style-type: none"> • Total percentage of methods that are covered • Number of covered methods, in green • Number of uncovered methods, in red |

Package level view displays a list of covered and not-covered classes for specific products and product builds. By clicking a class name in **Package** view you can drill down to view code-coverage information for the methods that are included in that class.

Package view displays the following attributes for each package in the selected product, across multiple rows:

| Attribute | Description |
|------------|---|
| Package | Package name |
| Statements | Total statements |
| Classes | <p>Histogram bar view that includes the following figures:</p> <ul style="list-style-type: none"> • Total percentage of classes that are covered • Number of covered classes, in green • Number of uncovered classes, in red |
| Methods | <p>Histogram bar view that includes the following figures:</p> <ul style="list-style-type: none"> • Total percentage of methods that are covered • Number of covered methods, in green • Number of uncovered methods, in red |

Class level view displays a list of covered and not-covered methods for specific products and product builds.

Class view displays the following attributes for each method, across multiple rows:

| Attribute | Description |
|------------|--|
| Name | Method name |
| Signature | Method signature |
| Statements | Total statements |
| Covered | <p>Covered status of method:</p> <ul style="list-style-type: none"> • True indicates that the method is covered • False indicates that the method is not covered |

 **Note:** When the page includes more elements than can be displayed at once without impacting response time, elements are displayed in increments. Use the navigation toolbar at the bottom of the page to browse through the elements.

Latest Builds and Build Versions

When you select a product in the navigation tree, the list of packages and classes with coverage information for the latest covered version, which implies the latest covered build for the version, is displayed automatically. When you select a product version in the navigation tree, coverage information for the latest covered build of the version is displayed automatically.

 **Note:** If you are running multiple applications against the same build, the coverage from all of the applications will be merged together.

 **Note:** Code analysis across a range of builds is not supported by Silk Central currently.

Generating Code-Change Impact Reports

To generate a code-change impact report:

1. In the menu, click **Projects > Project List**.
2. Select the project for which you want to analyze code-coverage data.
3. In the menu, click **Reports > Code Analysis**.
4. Click **Create Code Change Impact Report**. The **Select Classes for Report** dialog box appears.
5. Select a **Product** and **Version** if you want to change the pre-selected values.
6. In the **Filter** field, type criteria to filter the packages/namespaces. For example, entering the string `published` will only list packages/namespaces that contain the string `published` in their names.
7. Select a package from the **Packages/Namespaces** list.
Use **Ctrl+Click** or **Shift+Click** to select multiple packages/Namespaces.
The classes that are available in the selected package/namespace are displayed in the **Classes** list.
8. In the **Classes** list, select a class that you want to have included as a source in your report.
Use **Ctrl+Click** or **Shift+Click** to select multiple classes.
9. Click **Add**. The selected classes are added to the **Selected classes** list.
10. Repeat the preceding steps until you have added all required classes to the **Selected classes** list.
You can remove classes from the **Selected classes** list by selecting entries and clicking **Remove** or by clicking **Remove All**.
11. Select a report from the **Select report** list.
12. Click **OK** to generate the report.

Select Classes for Report Dialog Box

Reports > Code Analysis > Create Code Change Impact Report

The **Select Classes for Report** dialog box enables you to select class files to be included as sources in a *Code-Change Impact Report*. The dialog box includes the following elements:

| Element | Description |
|----------------|---|
| Product | Name of the product for which code analysis information is required. |
| Version | Version of the product for which code analysis information is required. |
| Filter | Enter criteria for filtering the packages. For example, if you type the string <code>published</code> , the <i>Code-Change Impact Report</i> displays only packages that contain the string “published” in their names. |

| Element | Description |
|----------------------------|--|
| Packages/Namespaces | Product packages/namespaces that are specified for coverage in the selected version. |
| Classes | All classes included in the selected package. |
| Add | Click to add the selected classes for code-coverage analysis. |
| Selected Classes | Displays the classes that are already selected for code-coverage analysis. |
| Remove | Click to remove the selected class from the Selected Classes list box. |
| Remove All | Click to remove all classes from the Selected Classes list box. |
| Select Report | Type of the <i>Code-Change Impact Report</i> . |

Viewing Code-Coverage Information for Packages/Namespaces

To view code-coverage information for a package/namespace:

1. In the menu, click **Projects > Project List**.
2. Select the project for which you want to view code-coverage information.
3. In the menu, click **Reports > Code Analysis**.
4. In the **Code Analysis** tree, expand the project node in the navigation tree to display the products that are available for the selected project.
5. Expand a product node to display the versions that are available for that product.
6. Expand a version node to display the builds that are available for that version.
7. Select a specific build. Code coverage information for the selected build displays on the **Details** page.



Note: To view code-analysis information for all products click **Show all products** on the toolbar. Products of other projects are then listed under the **Other Projects** node.

Default Reports

Context-Sensitive Reports

The areas **Requirements**, **Tests**, and **Executions** offer dynamically-generated lists of reports that are specific to each unit. Context-sensitive report lists are helpful because they offer report types that relate directly to your current activities.

Requirements Context-sensitive report lists in the **Requirements** tree offer all reports that take a requirement ID as an input parameter.

Tests Context-sensitive report lists in the **Tests** tree offers all reports that take a test ID as an input parameter.

Executions Context-sensitive report lists in the **Execution Plans** tree offer all reports that take an execution-plan ID as an input parameter. On the execution-plan **Runs** tab, context-sensitive report lists offer all reports that have the following configuration:

- **Result category** = Execution Plan
- **Selection criteria** = Execution Plan Run
- **Property** = ID

When you select a report from a context-sensitive report list, you are taken directly to that report's default tab in the **Reports** area. This default destination-tab behavior can be configured using each report's **Edit Report** dialog box.

There are two types of reports that appear in the context-sensitive report lists: reports that you have already accessed and reports that you have not yet accessed. Reports that you have accessed previously

appear above a line separator in the menu. These reports are listed chronologically with the most recently viewed report at the top of the list. Other default reports that are available, but have not yet been accessed, appear beneath the line separator.

In addition to the default-configured context-sensitive reports, you can configure new and existing reports to be included in the context-sensitive report list of each area. Context sensitivity is added to reports on a per-user, per-report basis only.

Accessing Context-Sensitive Execution Reports



Note: Reports must be enabled as context-sensitive reports to make them available in the **Executions** area.

To access a context-sensitive execution-plan report:

1. In the menu, click **Execution Planning**.
2. Right-click an execution plan in the **Execution Plans** tree and choose **Reports**.
3. Select a report from the **Reports** sub-menu.

You are taken to the **Parameters** tab of the selected report in the **Reports** area where the ID of the execution plan is pre-populated as a value.



Note: You can configure this destination-tab linking behavior by using each report's **Edit Report** dialog box.

4. Edit the parameters of the report as required.
5. Complete configuration of the report on the **Data, Report, or Chart** page.

Accessing Context-Sensitive Execution-Plan-Run Reports



Note: Reports must be enabled as context-sensitive reports to make them available in the **Executions** area.

To access a context-sensitive execution-plan-run report:

1. In the menu, click **Execution Planning > Details View**.
2. Click the **Runs** tab.
3. Right-click a run and choose **Reports**.
4. Select a report from the **Reports** sub-menu.

You are taken to the **Parameters** page of the selected report in the **Reports** area where the ID of the run is pre-populated as a value.



Note: You can configure this destination-tab linking behavior by using each report's **Edit Report** dialog box.

5. Edit the parameters of the report as required.
6. Complete configuration of the report on the **Data, Report, or Chart** page.

Accessing Context-Sensitive Requirements Reports



Note: Context-sensitive reports are available in the **Requirements** area only for those reports that accept a requirement ID as an input parameter.

To access a context-sensitive requirements report:

1. In the menu, click **Requirements > Details View**.
2. Right-click a requirement in the **Requirements** tree and choose **Reports**.
3. Select a report from the **Reports** sub-menu.

You are taken to the **Parameters** page of the selected report in the **Reports** area where the requirement's ID is pre-populated as a value.



Note: You can configure this destination-tab linking behavior by using each report's **Edit Report** dialog box.

4. Edit the parameters of the report as required.
5. Complete configuration of the report on the **Data, Report, or Chart** page.

Accessing Context-Sensitive Test Reports



Note: Context-sensitive reports are available in the **Tests** area only for those reports that accept a test ID as an input parameter.

To access a context-sensitive test report:

1. In the menu, click **Tests > Details View**.
2. Right-click a test, a folder, or a test container in the **Tests** tree or the **Grid View** and choose **Reports**.



Note: When multi-selecting tests in the **Grid View**, the context-sensitive reporting is disabled.

3. Select a report from the **Reports** sub-menu.

You are taken to the **Parameters** page of the selected report in the **Reports** unit where the ID of the test is pre-populated as a value.



Note: You can configure this destination-tab linking behavior by using each report's **Edit Report** dialog box.

4. Edit the parameters of the report as required.
5. Complete configuration of the report on the **Data, Report, or Chart** page.

Enabling Context-Sensitive Execution Reports

Explains how to enable execution-plan and execution-plan-run reports to be displayed in context-sensitive report lists.

To enable a simple report to be displayed in context-sensitive report lists in the **Execution** tree or the **Runs** page:

1. Create a new report.
For additional information, see *Creating New Reports*.
2. Select **Execution Plan** from the **Result category** list box.
3. Select the **Selection criteria** for the context-sensitive report.
4. Select **ID** from the **Property** list box.
5. Type a value in the **Value** field.
For example, the ID number of an existing execution plan or an existing execution-plan run.
6. Click **Finish**.

Enabling Advanced Context-Sensitive Execution Reports

To enable an advanced report to be displayed in context-sensitive report lists in the **Execution Plans** tree or on the **Runs** page:

1. Create a report that includes one of the following:
 - An execution-plan ID as an input parameter for the report to be displayed in the **Execution Plans** tree.
 - An execution-plan-run ID as an input parameter for the report to be displayed on the **Runs** page.For additional information, see *Creating New Reports* and *Writing Advanced Queries with SQL*.
2. To make an advanced query available in the context menu of the **Executions** area, insert the parameter name `execProp_Id_0` as input for `ExecDef_ID_pk_fk`.

For example, your report's SQL statement might have defined a hard-coded database-column value, such as `ExecDef_ID_pk_fk = 68`. To edit this report so that it receives column-name values dynamically, replace the static value of 68 with `${execProp_Id_0 | 68}`.



Note: For additional information about tables and column-name definitions, refer to the *Silk Central Database Model Schema*.

Enabling Context-Sensitive Requirement Reports

Explains how to enable requirements reports to be displayed in the context-sensitive report list.

To enable a simple report to be displayed in the context-sensitive report list of the **Requirements** area:

1. Create a new report.
For additional information, see *Creating New Reports*.
2. Select `Requirement` from the **Result category** list box.
3. Select the **Selection criteria** for the context-sensitive report.
4. Type a value in the **Value** field.
For example, the ID number of an existing requirement.

Enabling Advanced Context-Sensitive Requirement Reports

To enable an advanced report to be displayed in context-sensitive report lists in the **Requirements** area:

1. Create a report that includes a requirement ID as an input parameter.
For additional information, see *Creating New Reports* and *Writing Advanced Queries with SQL*.
2. To make an advanced query available in the context menu of the **Requirements** unit, insert the parameter name `reqProp_Id_0` as input for `Req_ID_pk_fk`.
For example, your report's SQL statement might have defined a hard-coded database-column value, such as `Req_ID_pk_fk = 68`. To edit this report so that it receives column-name values dynamically, replace the static value of 68 with `${reqProp_Id_0 | 68}`.



Note: For additional information about tables and column-name definitions, refer to the *Silk Central Database Model Schema*.

Enabling Context-Sensitive Test Reports

Explains how to enable test reports to be displayed in the context-sensitive report list.

To enable a simple report to be displayed in the context-sensitive report list of the **Tests** area:

1. Create a new report.
For additional information, see *Creating New Reports*.
2. Select `Test` from the **Result category** list box.
3. Select the **Selection criteria** for the context-sensitive report.
4. Type a value in the **Value** field.
For example, the ID number of an existing test.

Enabling Advanced Context-Sensitive Test Reports

To enable an advanced report to be displayed in context-sensitive report lists in the **Tests** area:

1. Create a report that includes a test ID, a test folder ID, or a test container ID as an input parameter.
For additional information, see *Creating New Reports* and *Writing Advanced Queries with SQL*.
2. To make an advanced query available in the context menu of the **Tests** area, insert the parameter name `tdProp_Id_0` as input for the test ID or `tfProp_Id` as input for the test folder ID or test container ID.

For example, your report's SQL statement might have defined a hard-coded database-column value, such as `TestDef_ID_pk_fk = 68`. To edit this report so that it receives column-name values dynamically, replace the static value of 68 with `#{tdProp_Id_0 | 68}`.



Note: For additional information about tables and column-name definitions, refer to the *Silk Central Database Model Schema*.

Requirements Reports

Requirements reports detail the status of functional requirements, for example compatibility requirements, GUI requirements, or feature requirements, which must be met during development.

Requirements may also relate to product management objectives such as reliability, scalability, and performance. The requirement-management reports help managers to determine if adequate test coverage is established to verify that system requirements are met during development. When a report references a requirement that includes HTML-formatted content, that content is rendered in the report.

Progress Reports

The following progress reports are available for the **Requirements** area:

| | |
|--|--|
| Requirements Coverage Across Builds 'X' and 'Y' | Represents a summary of all requirements by current requirement coverage. The returned group of requirements is restricted by the <code>Priority</code> parameter, which specifies the lowest requirement priority that is considered in the data. |
| Requirements Coverage Over the Past 'X' Days | Represents a trend in requirements coverage by considering overall requirements coverage over 'X' days. |
| Specific Requirements Coverage Over the Past 'X' Days | Represents a trend in requirements coverage by considering specific requirements coverage over 'X' days. |

Status Reports

The following status reports are available for the **Requirements** area:

| | | | | | | | |
|---|---|---------------|------------------|---------------|------------------|------------|---|
| Requirements Status Overview | Represents a grouped summary of all requirements by current requirement coverage. Coverage is expressed by the statuses <code>Passed</code> , <code>Failed</code> , <code>Not Executed</code> , and <code>Not Covered</code> . | | | | | | |
| Top-Level Requirement Coverage | Represents a listing of all top-level requirements. For each requirement the number of child requirements, that are covered and not-covered by tests, is displayed. | | | | | | |
| Status of Requirements with Priority Higher than 'X' | Represents a summary of all requirements by current requirement coverage. The returned group of requirements is restricted by the <code>Priority</code> parameter, which specifies the lowest requirement priority that is considered in the data. | | | | | | |
| Requirement Impact Analysis Report | Displays the test details grouped by the execution plan that is associated to the requirement. This report allows the user to gain insight into testing assets that may be impacted by a change to the requirement. The report contains the name, status, date/time of last execution, manual testers, planned time, and issues of each test. The following statuses are used: <table><tr><td>Passed</td><td>The test passed.</td></tr><tr><td>Failed</td><td>The test failed.</td></tr><tr><td>N/A</td><td>The test is associated to a requirement but is not submitted for execution.</td></tr></table> | Passed | The test passed. | Failed | The test failed. | N/A | The test is associated to a requirement but is not submitted for execution. |
| Passed | The test passed. | | | | | | |
| Failed | The test failed. | | | | | | |
| N/A | The test is associated to a requirement but is not submitted for execution. | | | | | | |

Not Executed The test is associated to a requirement and is submitted for execution but not executed yet.

Project Requirement Traceability Report Displays all requirements in a project along with their associated tests and issues. The following statuses are used:

Passed The test passed.

Failed The test failed.

N/A The test is associated to a requirement but is not submitted for execution.

Not Executed The test is associated to a requirement and is submitted for execution but not executed yet.

Document Reports

The following document reports are available for requirements:

All Requirements All requirements are represented with full requirement information.

Requirement with Child Requirements The selected requirement is shown with its requirement ID. Full details regarding the child requirements of the requirement are displayed.

All Related Issues Report

The **All Related Issues** report provides a detailed list of all issues related to the assigned tests for a requirement, and explains the relationship between requirements, the assigned tests, and issues that have occurred. The input parameter for the **All Related Issues** report is the identifier of the requirement.

Test Reports

This section explains the test-related reports that ship with Silk Central. Test reports give you an overview of the progress of your tests and the status of defects over a period of time or over a range of builds.

Progress Reports

The following progress reports are available for the **Tests** area:

Test Progress Across Builds 'X' and 'Y' Represents a trend in test progress resulting from viewing test statuses in context with builds. The user must specify a build range consisting of a start- and an end-build.

Manual Test Coverage since Build <x> (cumulative) Represents a trend in manual test progress, excluding all kind of automated tests, resulting from viewing test statuses in context with builds. The user must specify a starting build number which is used as basis for calculating a cumulative progress.

Automated Test Coverage for Builds (non-cumulative) Represents the automated test coverage resulting from viewing test statuses in context with builds. The user must specify a version and a product and the report shows the test coverage that could be achieved with the automated tests per build.

Progress of Tests in Specific Folder Over the Past <x> Days Represents a trend in test progress resulting from viewing test statuses for the test tree node with the given ID and for the last <x> number of days.

Test Progress Over the Past 'X' Days Represents a trend in test progress by considering test statuses over the past 'X' days.

| | |
|--|--|
| Test Progress this Month | Represents a trend in test progress resulting from viewing test statuses for the current month. |
| Percentage Testing Success Over the Past 'X' Days (per component) | Represents a percentage listing of successful tests per component over the past 'X' days; assists in identifying the components in the environment that are most critical. |
| Tests Created in the Past 'X' Days (per component) | Represents a listing of new tests per component over the past 'X' days. Assists in identifying components that lack testing activity. |

Manual Test Reports

The following manual-test reports are available for the **Tests** area:

| | |
|--|--|
| Historic Planned vs. Actual Execution Time (per user) | Represents an overview of planned and actually required execution time for completed manual tests per user over a specific period of time. |
| Planned vs. Actual Execution Time (status per user) | Represents progress in terms of planned vs. actual hours of currently pending manual tests per user. Manual tests are only considered if test results have been entered by the user and are assigned to the user who enters the results. |
| Manual Test Result Document | An easily printable manual test case report for the latest results of all tests assigned to the specified execution plan. |
| Manual Test Result Document (Microsoft Word) | An easily printable manual test case report for the latest results of all tests assigned to the specified execution plan in a Microsoft Word document. |
| Manual Test Results by Execution Plan | An easily printable manual test case report for the latest results of all tests assigned to the specified execution plan folder or testing cycle. |
| Planned vs. Actual Execution Time of Manual Tests (Summary) | Represents an overview of the deviation between planned and actual time for execution of manual tests, viewed on a daily basis. |

Status Reports

The following status reports are available for the **Tests** area:

| | |
|--|--|
| Tests per Component | Represents an overview of coverage of components by test; makes it easier to see where testing activity is needed. |
| Implemented Tests (per component) | Represents an overview of coverage of components by tests that have the <i>Implemented</i> attribute set to <i>Yes</i> . |
| Failed Tests (per component) | Represents an overview of failed tests per component; makes it easier to identify the most critical components in the environment. |
| Tests (per test container) | Represents a success rate for each test container by listing the number of passed tests. |
| Quality Goals Planning Report | Represents the number of tests in each goal criteria along with the planned time rolled up to the goal criteria. |



Note: The planned time value is only for manual tests. Automated tests do not have planned time.

This report is split three ways for each quality goal defined in a project:

- **Testing Scope** - compares the total number of tests available against the total number or required tests for each item in a quality goal.
- **Required Planned Time** - compares the required planned time among the items in the quality goals.

- **Value table** - lists all of the quality goal items along with related metrics: **Tests Available**, **Goal (%)**, **Tests Required**, and **Planned Time for Manual Tests**.



Note: For a multi select attribute/property, the test will be counted for each matching occurrence.

Quality Goals Execution Report

Visually displays the number of tests executed for each item of each goal. This report is split into two sections for each quality goal defined in a project:

- **Execution Status Graph** - lists each quality goal item with a color-coded execution status and summary value.
- **Value table** - lists all of the quality goal items as well as their execution metrics: **Passed**, **Failed**, **Not Executed**, **N/A**, **Total**, **Tests to Meet Goal**, and **Execution % to Goal**.

Test Status Overview

Represents a status overview of all tests, structured by the statuses `Passed`, `Failed`, `Not Executed`, and `N/A`.

Test Status Overview (per test container)

Represents a status overview of all tests contained in a specific test container, structured by the statuses `Passed`, `Failed`, `Not Executed`, and `N/A`.

Test Run Comparison Report

Test run-comparison reports compare the last two runs of a test by providing detailed information about context, status, errors, warnings, duration, properties, attributes, parameters, and success conditions.



Note: Run-comparison reports are not suitable for the comparison of manual tests to automated tests. When the name of the report includes (*Last Two Runs*), you can compare only the last two runs of the execution plan or test.

Baseline Tests Comparison Report

The **Baseline Tests Comparison** report compares a baselined project with the baseline and displays the number of changed, deleted, and created tests.

The input parameters for the **Baseline Tests Comparison** report are the identifiers of the project and the baseline.

Execution Planning Reports

This section explains the execution planning reports that ship with Silk Central. To ease the assessment of results, execution planning reports give you a detailed overview of the progress of your test executions and the status of defects, over a period of time, or over a range of builds.

Execution Plan Run Comparison Reports

Execution-plan run-comparison reports present an overview of the comparison between two execution-plan runs.



Note: Run-comparison reports are not suitable for the comparison of manual tests to automated tests. When the name of the report includes (*Last Two Runs*), you can compare only the last two runs of the execution plan or test.

The following reports compare two execution-plan runs:

Execution Plan Run Comparison Report

The default execution-plan run-comparison report that compares two execution-plan runs.

Execution Plan Run Comparison Report – Changed Status

Compares only those tests of two execution-plan runs, that changed their statuses.

Execution Plan Run Comparison Report – Failed in Newer Run Compares only the failed tests of two execution-plan runs.

The execution-plan run-comparison report provides the following information:

- Changes to the status of the execution plans
- Number of errors
- Number of warnings
- Context in which the execution plans were executed
- Execution duration of the assigned tests



Note: When the status of an assigned test changes to `Failed` between compared runs, the test is marked red. When the status of an assigned test changes to `Passed` between compared runs, the test is marked green.

Execution Plan Run Errors Report

The execution plan **Run Errors** report provides a detailed list of all tests that did not pass an execution-plan run and the reason they did not pass.

All errors that occurred during the execution-plan run are listed in this report. The user can quickly assess results and easily identify any unwanted effects in the execution-plan run. The input parameter for an execution plan **Run Errors** report is the identifier of the execution-plan run.

Tests Not Passed Report

The **Tests Not Passed** report presents an overview of all not-passed tests of a folder or project in the **Executions** area.

You can use this report for error analysis, for example during configuration testing. The input parameter for a **Tests Not Passed** report is the identifier of the configuration suite, folder, or project.

Tests Not Passed per Execution Plan Report

The **Tests Not Passed per Execution Plan** report presents an overview of all not-passed tests per configuration of a configuration suite, or per execution plan of a folder or project in the **Executions** area.

You can use this report for error analysis, for example during configuration testing. The input parameter for a **Tests Not Passed per Execution Plan** report is the identifier of the configuration suite, folder, or project.

Execution Status Overview Report

The **Execution Status Overview** report presents an overview of the last run status of all tests assigned to the execution plans of the selected node.

The input parameter for an **Execution Status Overview** report is the identifier of the node in the **Execution Plans** tree.

Issues Per Component Report

The **Issues Per Component** report offers an overview of all issues related to each component.

In addition to offering basic issue tracking, this report assists in monitoring the overall issue trend for each component.

Code Coverage Reports

This section explains the code coverage reports that ship with Silk Central. Code coverage reports offer a detailed overview of your product's code coverage over a period of time or range of builds.

Code-Change Impact Reports

The code-change impact reports enable you to perform testing-impact analysis, effort analysis, and risk analysis. You can select classes of interest and, by applying report templates, generate reports that help

you determine the test impact that changing the selected classes will result in. For selected classes, you can choose from report templates that analyze the test impact of proposed code changes.

The **Code Change Impact - Tests** report identifies the tests that need to be re-run following code changes. The coverage index displays the methods covered by the test for the specified classes / total methods of specified classes.

The **Code Change Impact - Execution Plans** report identifies the execution plans that need to be re-run following code changes.

Use Cases for Reports

The following typical code-change impact issues can be addressed with code-change impact reports:

Testing impact analysis

You want to know which tests you should run as a result of a specific change to the code.

- Select a particular class.
- Select and execute the **Code Change Impact - Tests** report.
- Observe the list of tests that cover the classes that were touched in this version.

Effort analysis

You want to know how many hours of automated and manual testing will be required to properly cover a particular set of changes to the code.

- Select a particular class.
- Select and execute the **Code Change Impact - Execution Plans** report.
- Observe the required time (cost) for automated and manual tests.

Code Coverage Trend Report

Shows the improvement trend of code coverage for methods, classes, and packages for a product over a selected range of builds.

Code Coverage Trend Graph

Shows the overall percentage of code coverage for the selected product over the selected range of builds. Code coverage for specific packages/namespaces, classes, and methods is displayed individually.

Code Coverage Trend Details

Displays the information from the *Code Coverage Trend Graph* in a tabular format.

Method Coverage Comparison Report

Compares the method coverage for all included packages/namespaces across two product builds.

Use the `Threshold` parameter to specify the minimum amount of change that results in a package/namespace appearing in the report. Packages/namespaces with a smaller percentage of change are not shown in the report. The threshold range is from 0 to 100 percent.

Performance Trend Reports

Performance trend reports show the evolution of the application under test's performance over a specified period of time.

The input data for the performance reports is provided by Silk Performer load tests.

Average Page-Time Trend Report

Shows the page times per page for all tests executed for the specified test within the specified time range. The performance trend of the page times for the tested pages is shown in a graph.

Input Parameters

The input parameters for an **Average Page-Time Trend** report are:

| Parameter | Description |
|---|--|
| Date From (DD-MON-YYYY) | Starting date for the time range. For example 06-DEC-2008. |
| Date To (DD-MON-YYYY) | End date for the time range. For example 16-JAN-2009. |
| Exclude Runs with more than <nnn> Errors | Runs that generate more errors than specified here are not included in the report. Use this setting to avoid that outliers skew the trend curve. |
| Maximum Value for y-Axis | Limits the y-axis of the graph to the specified value. Transaction busy-times that exceed this value are cut off at the top. This setting is useful to prevent the flattening of lines caused by outliers. |
| Measure Filter | Shown measures are limited to those including the specified string in their name. This field has to be filled out. To display all available measures, set the measure filter to %. For example, to show only measures that include the word "unit" at any position in their names, set the measure filter to %unit%. |
| Test ID | Identifier of the test for which you want to view the report. |

Average Transaction Busy-Time Trend Report

Shows the transaction busy time per transaction for all tests executed for the specified test within the specified time range. The performance trends of the transaction busy-times for the tested transaction are displayed in trend charts.

Input Parameters

The input parameters for an **Average Transaction Busy-Time Trend** report are:

| Parameter | Description |
|---|--|
| Date From (DD-MON-YYYY) | Starting date for the time range. For example 06-DEC-2008. |
| Date To (DD-MON-YYYY) | End date for the time range. For example 16-JAN-2009. |
| Exclude Runs with more than <nnn> Errors | Runs that generate more errors than specified here are not included in the report. Use this setting to avoid that outliers skew the trend curve. |
| Maximum Value for y-Axis | Limits the y-axis of the graph to the specified value. Transaction busy-times that exceed this value are cut off at the top. This setting is useful to prevent the flattening of lines caused by outliers. |
| Test ID | Identifier of the test for which you want to view the report. |
| Transaction Filter | Shown transactions are limited to those including the specified string in their name. This field has to be filled out. To display all available transactions, set the transaction filter to %. For example, to show only transactions that include the word "unit" at any position in their names, set the transaction filter to %unit%. |

Overall Page-Time Trend Report

Shows overall page times, aggregated over all user types, for all tests executed for the specified test within the specified time range. The performance trend of the page times for the tested page is shown in a graph.

Input Parameters

The input parameters for an **Overall Page-Time Trend** report are:

| Parameter | Description |
|---|--|
| Date From (DD-MON-YYYY) | Starting date for the time range. For example 06-DEC-2008. |
| Date To (DD-MON-YYYY) | End date for the time range. For example 16-JAN-2009. |
| Exclude Runs with more than <nnn> Errors | Runs that generate more errors than specified here are not included in the report. Use this setting to avoid that outliers skew the trend curve. |
| Maximum Value for y-Axis | Limits the y-axis of the graph to the specified value. Transaction busy-times that exceed this value are cut off at the top. This setting is useful to prevent the flattening of lines caused by outliers. |
| Test ID | Identifier of the test for which you want to view the report. |

Overall Transaction Busy-Time Trend Report

Shows overall transaction busy-time, aggregated over all user types, for all tests executed for the specified test within the specified time range. The performance trend of the transaction busy-times for the tested transaction is displayed in a trend chart.

Input Parameters

The input parameters for an **Overall Transaction Busy-Time Trend** report are:

| Parameter | Description |
|---|--|
| Date From (DD-MON-YYYY) | Starting date for the time range. For example 06-DEC-2008. |
| Date To (DD-MON-YYYY) | End date for the time range. For example 16-JAN-2009. |
| Exclude Runs with more than <nnn> Errors | Runs that generate more errors than specified here are not included in the report. Use this setting to avoid that outliers skew the trend curve. |
| Maximum Value for y-Axis | Limits the y-axis of the graph to the specified value. Transaction busy-times that exceed this value are cut off at the top. This setting is useful to prevent the flattening of lines caused by outliers. |
| Test ID | Identifier of the test for which you want to view the report. |

Custom Measure Trend Report

Shows the average, minimum, and maximum values of the defined measure or measures for all tests executed for the specified test within the specified time range. The performance trend of the values for each tested measure is shown in a graph.

Input Parameters

The input parameters for a **Custom Measure Trend** report are:

| Parameter | Description |
|--------------------------------|--|
| Date From (DD-MON-YYYY) | Starting date for the time range. For example 06-DEC-2008. |
| Date To (DD-MON-YYYY) | End date for the time range. For example 16-JAN-2009. |

| Parameter | Description |
|---|--|
| Exclude Runs with more than <nnn> Errors | Runs that generate more errors than specified here are not included in the report. Use this setting to avoid that outliers skew the trend curve. |
| Maximum Value for y-Axis | Limits the y-axis of the graph to the specified value. Measures that exceed this value are cut off at the top. This setting is useful to prevent the flattening of lines caused by outliers. |
| Measure Name | Name of the custom measure for which you want to view the report. For example <i>CreateTestDefinition</i> . |
| Measure Type | Type of the custom measure. For example <code>Transaction(BusyTime)[s]</code> . |
| Test ID | Identifier of the test for which you want to view the report. |

Project Overview Report

Silk Central > Tracking > Project Overview Report

The *Project Overview Report* contains a high-level overview of the status of the selected project.

Administration

Once you have completed the initial configuration of Silk Central (instance configuration), this section will guide you through the steps required to set up user accounts, projects, locations, execution servers, and more. These tasks must be performed by an administrator.

User Roles and Permissions

When working with Silk Central, tasks are assigned to designated groups of users who have access to assigned projects. Within groups, users are granted specific roles within those projects. User permissions are configured based on user role type and group membership. This topic defines each permission type and details the specific permissions that are associated with each user role.

Each user account can belong to one or multiple groups. A group specifies which roles a user has within that group. Groups are assigned to projects. So the permissions that each individual user has are derived from the group/role assignments that have been defined for them. Defined permissions apply only to the projects that are assigned to the groups in which each user has a group/role assignment.

User Roles

The following user roles are predefined in Silk Central:

- SuperUser
- Administrator
- Project Manager
- Test Manager
- Tester
- Analyst
- Reporter
- Role Assignment Manager

These roles cannot be modified or deleted. They can however be copied and thereby used as the basis for customized roles.

SuperUser

The SuperUser role is a special role that is granted all privileges across Silk Central.

Administrator

Administrator tasks include the configuring of application-, Web-, and chart-server locations; setting up and maintaining repositories and notification settings; creating accounts; configuring locations and execution servers, and others.

Administrators are granted all privileges across Silk Central and Issue Manager.

Project Manager

Project Managers maintain the projects for which they are responsible. Project Managers do not have write access to the Silk Central **Administration** unit. Project Managers can only access the projects to which they have been assigned as Project Managers, where they have full write access to all project-related features. Project Managers also have all Issue Manager permissions for projects that are assigned to them.

Test Manager

Test Manager responsibilities include the planning and execution of tests, including the deletion of tests. Test Managers also have full access to libraries and full read access to the **Requirements** area in Silk Central.

Tester

The Tester role relates to Silk Central privileges. The Tester's tasks include the planning and execution of tests—though Testers cannot delete tests. Testers also have full read access to the **Requirements** area, and can view, create, and edit all objects in libraries.

Analyst

Analysts analyze the results of projects that have been assigned to them. They cannot modify project settings or schedules and have read-only privileges.

Reporter

In addition to having all the rights of Analysts, Reporters additionally have the right to edit and delete reports in *Advanced mode*. Advanced mode allows reporters to enter, modify, and delete SQL statements for advanced reports. For details on advanced reports, refer to the Silk Central Help.

Role Assignment Manager

Role Assignment Managers have the sole permission to assign/unassign roles to/from existing users of a group, and only roles that they themselves are assigned to. They cannot add or remove members from a group, and they can only change role assignments within groups in which they themselves are defined as Role Assignment Manager.

Adding User Roles

To add a user role:

1. In the menu, click **Administration > User Management**.
2. Click the **Roles** tab.
3. Click **New Role**.

The **New Role** page displays.

4. Type a **Name** for the new role.
5. *Optional:* Type a **Description** for the role.
6. In the **Permission Settings** list, check the **Allow** field for all permissions you want to grant to this role.



Note: Checking a top-level parent task automatically checks all child tasks of that parent. When some but not all child tasks of a parent task are selected, the parent task is checked with a grayed-out check mark, indicating partial permissions in that area.

7. Click **Save** to save your permission settings for this role.

Editing User Roles



Note: Predefined user roles cannot be edited. Custom user roles can be edited.

To edit a user role:

1. In the menu, click **Administration > User Management**.
2. Click the **Roles** tab.
3. Click the name of the role that you want to edit in the **Roles** list.

The **Edit Role** page displays.

4. Edit the **Name** of the role as required.
5. Edit the **Description** of the role as required.
6. In the **Permission Settings** list, check the **Allow** check boxes of all permissions that you want to grant to this role. Uncheck any selected permissions that are not to be granted to this role.

 **Note:** Checking a top-level parent task automatically checks all child tasks of that parent. When some but not all child tasks of a parent task are selected, the parent task is checked with a grayed-out check mark, indicating partial permissions in that area.

7. Click **Save** to save your permission settings for this role.

Copying User Roles

 **Tip:** Copying existing user roles is the first step in creating a custom user role. After copying an existing role, rename it and edit its permissions to meet your needs.

To copy a user role:

1. In the menu, click **Administration > User Management**.
2. Click the **Roles** tab.
3. In the **Actions** column of the user role that you want to copy, click **Duplicate Role**.
The copy of the role then displays in the list of user roles where you can rename it and customize it as required.

Deleting User Roles

 **Note:** Predefined user roles cannot be deleted. Custom user roles can be deleted.

To delete a user role:

1. In the menu, click **Administration > User Management**.
2. Click the **Roles** tab.
3. In the **Actions** column of the user role that you want to remove, click **Delete**. A confirmation dialog box displays.
4. Click **Yes** to confirm the operation; click **No** to abort. If you choose **Yes**, you will be returned to the list of user roles where the deleted role will no longer be listed.

Roles Settings Page

Administration > User Management > Roles

The **Roles Settings** page is used to configure user roles. The page displays the following items:

| Item | Description |
|-----------------------------|---|
| Actions | Click  to duplicate a role and use it as the basis for a new, custom role. Click  to delete a user role. |
| Name | The name of the user role as it displays in the GUI. |
| Allow New Assignment | Click to allow or prevent an existing user role from accepting new user assignments. This is useful when a user role has been discontinued while some user accounts still retain the role. |
| Description | Description of the user role. |
| Created On | Date the role was created. |
| Created By | User who created the role. |

Permission Definitions

To display the permissions in Silk Central: In the menu, click **Administration > User Management**. Click the **Roles** tab. Click on a role in the grid. The permissions for that role display.

This section explains the permissions that govern user ability to perform tasks and access secure areas within Silk Central. There is a separate list for each permission category.



Note: Permissions for predefined roles cannot be edited.

Requirements Permissions

The following permissions are available for requirements:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|---|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View requirements | | X | | X | X | X | X |
| Manage requirements | | X | | | | | |
| Delete requirements | | X | | | | | |
| Manage requirements management integrations | | X | | | | X | |
| Delete requirements management integrations | | X | | | | | |

Libraries Permissions

The following permissions are available for libraries:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|------------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View libraries | | X | | X | X | X | X |
| Manage libraries | | | | | X | X | |

| Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| | | | | | X | |

Tests Permissions

The following permissions are available for tests:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|------------------------------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View tests | | X | | X | X | X | X |
| Manage tests | | | | | X | X | |
| Delete tests | | | | | | X | |
| Approve tests | | | | | | X | |
| Manage source control integrations | | X | | | | X | |
| Delete source control integrations | | X | | | | X | |
| Import/update tests from Excel | | | | | X | X | |

Executions Permissions

The following permissions are available for executions:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|----------------------------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View executions | | X | | X | X | X | X |
| Manage executions | | | | | X | X | |
| Delete executions | | | | | | X | |
| Start executions | | | | | X | X | |
| Finish executions | | | | | X | X | |
| Execute tests | | | | | X | X | |
| Synchronize manual test runs | | | | | X | X | |
| Delete execution plan runs | | | | | X | X | |
| Change test run status and build | | | | | X | X | |

Manual Execution Planning Permissions

The following permissions are available for manual execution planning:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|--|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View testing cycles and configurations | | X | | X | X | X | X |
| Manage testing cycles and configurations | | | | | | X | |
| Delete testing cycles and configurations | | | | | | X | |

Issues Permissions

The following permissions are available for issues:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|------------------------------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| Manage issue tracking integrations | | X | | | | X | |
| Delete issue tracking integrations | | X | | | | X | |

Quality Goals Permissions

The following permissions are available for quality goals:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|----------------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View quality goals | | X | | X | X | X | X |
| Manage quality goals | | X | | | | X | |
| Delete quality goals | | X | | | | X | |

Reports Permissions

The following permissions are available for reports:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|------------------------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View reports | | X | | X | X | X | X |
| Manage reports | | X | | | X | X | X |
| Manage advanced reports | | | | | | | X |
| Delete reports | | X | | | X | X | X |
| Manage shared global reports | | | | | | | X |
| Delete shared global reports | | | | | | | X |

Shared Global Report Permissions

Because shared global reports are available in all projects for all users, user roles that are able to update and delete them require special permissions: **Manage shared global reports** and **Delete shared global reports**. These permissions also apply to report templates that are used in shared global reports. Non-shared global reports and report templates can be updated and deleted by user roles with the standard report permissions for updating and deleting reports. This implies the following special scenarios:

- The option **Share this report with other users** is disabled in global reports and folders if the user does not have the **Manage shared global reports** permission
- When pasting a copied shared report into the **Global Reports** tree without the **Manage shared global reports** permission, the pasted report is set to non-shared

Projects Permissions

The following permissions are available for projects:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|-----------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| Manage projects | | X | | | | X | |
| Delete projects | | X | | | | X | |

Project Settings Permissions

The following permissions are available for project settings:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|---------------------------------------|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View project settings | | X | | X | X | X | X |
| Manage project settings | | X | | | X | X | |
| Delete project settings | | X | | | | X | |
| Manage filters | | X | | | X | X | |
| Delete filters | | X | | | X | X | |
| Modify private filters of other users | | X | | | | X | |
| Delete private filters of other users | | X | | | | X | |
| Modify Office import mapping files | | X | | | | X | |
| Delete Office Import Mapping Files | | X | | | | X | |
| Manage status reasons | | | | | X | X | |
| Delete status reasons | | | | | | X | |

Administration Permissions

The following permissions are available for administration:

| | Administrator | Project Manager | Role Assignment Manager | Analyst | Tester | Test Manager | Reporter |
|---|---------------|-----------------|-------------------------|---------|--------|--------------|----------|
| View system settings | X | X | | | | X | X |
| Manage system settings | X | | | | | | |
| View administration settings | X | X | X | | | X | X |
| Manage administration settings | X | | | | | | |
| Manage role assignments of users | X | | X | | | | |
| Delete administration settings | X | | | | | | |
| View log files | X | X | | | | X | X |
| Delete log files | X | | | | | | |
| View audit logs | X | | | | | | |
| Manage execution servers | X | | | | | | |
| Delete execution servers | X | | | | | | |
| Activate and deactivate execution servers | X | | | | | | |
| View products, versions and builds | X | X | | | | X | X |
| Manage products, versions and builds | X | | | | | | |
| Delete products, versions and builds | X | | | | | | |

Dashboard Panel Permissions

To view or edit the content of a certain dashboard panel, you need the following permissions:

| Panel | Required permission |
|--|---|
| Assigned Issues | View tests. |
| Custom Information | Manage projects required for editing. No permission required for viewing. |
| Introduction | No permission required. |
| Issue Life Cycle | View project settings. |
| Issues Created per Tester | View tests or view executions. |
| Manual Tests Assigned to Me | Execute tests. |
| Offline Manual Testing Overview | View tests or view executions. |
| Planned vs. Actual Execution Time | View testing cycles and configurations. |

| Panel | Required permission |
|--|---|
| Quality Goal Progress | View quality goals. |
| Report Subscription | No permission required. |
| Requirements Coverage Status | View requirements. |
| Silk Performer Performance Trends | View tests or view executions. |
| Status of Currently Running Manual Tests | View tests or view executions. |
| Testbook | No permission required. |
| Testing Cycle Progress | View testing cycles and configurations. |
| Testing Cycle Result Summary | View testing cycles and configurations. |
| Testing Cycle Timeline | View testing cycles and configurations. |
| Testing Progress Across Testing Cycles | View testing cycles and configurations. |
| Volatile Tests | View tests or view executions. |

User Accounts and Groups

A user account must be created for each user working with Silk Central. One or more groups of users are assigned to specific projects. Only with a user account, a user role, and a group assignment can a user work with a Silk Central project.

Maintaining User Accounts

User accounts track login data and configuration settings for individual users. They also enable user login. User accounts are typically assigned to group accounts with one or more specific user roles for specific projects. The SuperUser is the only user role that can, among other things, configure the application-, Web-, and chart server locations; and set up and maintain repositories and notification settings.



Caution: Because the *SuperUser* account `admin` has all administrative privileges, you should immediately create a new password for this user to prevent unlimited access to these privileges. For more information on changing the password, see **Changing the Password of the System Administrator Account**.

Adding User Accounts

To add a user account:

1. In the menu, click **Administration > User Management**.
2. Click the **Accounts** tab.
The page displays all available user accounts. When you access this page for the first time, the *SuperUser* account `admin` is the only user listed.
3. Click **New User**. The **Add new user account** page displays.
4. Type a username and password for the user. Type the password a second time to confirm it.
5. Check the **Mixed mode authentication (LDAP)** check box to enable both LDAP and local-credential based authentication.
6. Set the login to **Locked** if you want to prevent the user from logging in.
7. Type the user's first name, last name and email address.

8. Type the user's local time zone and select a date format, a short date format, and the first day of the week.
9. Type the **Page refresh time** in seconds, the **CSV separator string**, and a **Default execution server**.
10. Select a group and role definition from the respective list boxes.
11. Click **Add Assignment** to add the group and role combination to the user account.
12. Repeat the previous two steps to assign all desired group and role combinations to the user account.
13. To remove a group and role combination from the current user account, click the **Delete** icon in the **Actions** column.
14. Click **Save** to save your settings.

Editing User Accounts

Once a user account is set up you may edit any of the parameters, except the **Login** name.



Note: Changes to a user account become active upon the next login of the changed user account. Please notify the user to logout and login again.

To edit a user account:

1. In the menu, click **Administration > User Management**.
2. Click the **Accounts** tab.
The page displays all available user accounts. When you access this page for the first time, the *SuperUser* account `admin` is the only user listed.
3. Click the **Login** name of the user account that you want to edit. The **Configure existing user** page displays.
4. Edit the password of the user as required. Type the password a second time to confirm it.
5. Check the **Mixed mode authentication (LDAP)** check box to enable both LDAP and local-credential based authentication.
6. Edit other user settings as required.
7. Select a group and role definition from the respective list boxes.
8. Click **Add Assignment** to add the group and role combination to the user account.
9. Repeat the previous two steps to assign all desired group and role combinations to the user account.
10. To remove a group and role combination from the current user account, click **Delete** in the **Actions** column.
11. Click **Save** to save your settings.

Deleting User Accounts



Caution: Deleting a user account is not reversible. You may lock a user account instead, if you want to temporarily make an account unavailable. For additional information about locking user accounts, see *Editing User Accounts*.

To delete a user account:

1. In the menu, click **Administration > User Management**.
2. Click the **Accounts** tab.
The page displays all available user accounts. When you access this page for the first time, the *SuperUser* account `admin` is the only user listed.
3. In the **Actions** column of the user account you want to remove, click **Delete**. A confirmation dialog box displays.
4. Click **Yes** to confirm the operation; click **No** to abort. If you choose **Yes**, you will be returned to the list of user accounts where the deleted account will no longer be listed.

User Settings Page

Administration > User Management > Accounts > New/Edit User

Use the **User Settings** page to configure user accounts. User account settings are closely related to group account settings.

You can click on the name of the user in the menu to access the **User Settings** page for the logged-in user.

 **Note:** You must define at least one group and role assignment to save a user account.

| Login Data Item | Description |
|---|--|
| Login | The username to be stored in the Silk Central repository. If you check Mixed mode authentication (LDAP) below, the entered username must match the defined LDAP username. |
| Password | Enter a valid password for the Login that you entered. This password is not related to the LDAP password. |
| Confirm password | Enter the password again to confirm it. |
| Mixed mode authentication (LDAP) | Check this check box to enable both LDAP and local-credential based authentication. If an LDAP server exists, not checking this check box results in LDAP-only authentication. |
| Locked | Check this check box if you want to prevent the user from logging in with the given credentials. This makes the user account inactive. |

| General Data Item | Description |
|---------------------------------|--|
| First name | Type the user's first name. This information does not affect the behavior of Silk Central; it simply tracks user contact information. |
| Last name | Type the user's last name. This information does not affect the behavior of Silk Central; it simply tracks user contact information. |
| Email | Type the user's email address. This information is used for notification purposes. |
| Theme | Select a Silk Central version to set the respective color scheme in which Silk Central's UI will be displayed for the selected user. |
| Time zone | The user's local time zone. Time zone information is used to display times and dates in the user's local time zone. |
| Date format | The selected date format is presented to the user in lists, reports, and in the calendar whenever Silk Central displays a long date format. |
| Short date format | The selected date format is presented to the user in lists, reports, and in the calendar whenever Silk Central displays a short date format. |
| First day of week | The first day of the week determines the weekly view in reports. |
| CSV separator string | This string is used as a row separator for the user's downloaded CSV-files. Reports can be downloaded as CSV-files. |
| Default Execution Server | The default execution server is used for trial runs of automated tests. |

| Group and Role Assignments Item | Description |
|---|--|
| Group and Role Assignments table | Lists all existing user group/user role assignments of the user. You can also delete group and role assignments by clicking  next to the assignment you want to remove. |
| Group | Select a group to which the user is to be assigned. This list box lists the user groups that have been defined by a Silk Central administrator. Groups that were imported from an LDAP server are only listed, if the user is assigned to them. |
| User role | Select the user role with which the user is to be assigned to the selected group. The list is populated with the predefined system roles and the custom user roles. |
| Add | Click this button to create a new user group/user role assignment with the group and user role you selected. |

Web-Service Token

This area enables to generate a web-service token, which you can use to authenticate to Silk Central through a web service.

| Item | Description |
|--------------------------------|---|
| Generate Token | Generates a new web-service token. |
| Copy Token to Clipboard | Copies the web-service token to the clipboard. This button is only enabled if a token has already been generated. |
| Regenerate Token | Re-generates a new web-service token. This button is only available if a token has already been generated. |
| Delete Token | Deletes a web-service token. This button is only enabled if a token has already been generated. |

Maintaining Groups

Groups define access to specific projects. Each user can be associated with one or more groups from which they inherit the access rights to the projects that are defined for the selected group.

 **Note:** Users can be added to groups with multiple roles, allowing advanced user permission configuration.

Adding Groups

To add a group:

1. In the menu, click **Administration > User Management**.
2. Click the **Groups** tab.
3. Click **New Group**.
4. In the **Group name** field, type a group name for the new group.
5. In the **Description** field, enter a description for the new group.
6. Select a user with a role assignment from the respective list boxes, then click **Add Selection** to add the user and role combination to the new group.

 **Note:** Any user roles that have been defined as not accepting new user assignments are not displayed in this list. These settings are controlled through the **Allow New Assignment** buttons at **Administration > User Management > Roles** .

For groups that were imported from an LDAP server, additional role assignments can be added to the users and role assignments can also be deleted, but at least one role assignment per user must remain. Removing users from an LDAP-imported group can only be done on the LDAP server itself.

7. Repeat the previous step to assign all desired user and role combinations to the group.



Tip: It may be useful to assign the role **Role Assignment Manager** to one or more users of the group. Role Assignment Managers have the sole permission to assign/unassign roles to/from existing users of a group, and only roles that they themselves are assigned to. They cannot add or remove members from a group, and they can only change role assignments within groups in which they themselves are defined as Role Assignment Manager.

8. To remove a user and role combination from the current group, click **X** in the **Actions** column.
9. In the **Project Assignment(s)** section you can assign any existing projects to this group.
10. Click **Save**.

Importing Groups from LDAP Servers

Import groups from LDAP servers and have your Silk Central users be created automatically.

Before you can import groups from LDAP servers, an LDAP server must be configured. For detailed information, see *Adding LDAP Servers*.

To import LDAP groups and their associated users into Silk Central:

1. In the menu, click **Administration > User Management**.
2. Click the **Groups** tab.
3. Click **Import LDAP Group**. The **Import LDAP Group** dialog box appears.
4. If your LDAP server configuration has been set up to use a domain, select an **LDAP Domain** if the imported users should log in to Silk Central with the pattern `<domain>\<username>`. Select the default domain if the imported users should log in to Silk Central without entering a domain name.
5. Select the **LDAP Group** that you want to import.
6. Select the **Silk Central Default Role**. All users of the selected **LDAP Group** will be set up with this role.
7. Click **OK**.

After importing a group from LDAP, it is recommended to check the group's settings, specifically the project assignments. Note that no users can be assigned to or removed from an imported LDAP group, as this assignment is strictly handled through the LDAP server. However, you can modify the individual user's roles.



Tip: It may be useful to assign the role **Role Assignment Manager** to one or more users of the group. Role Assignment Managers have the sole permission to assign/unassign roles to/from existing users of a group, and only roles that they themselves are assigned to. They cannot add or remove members from a group, and they can only change role assignments within groups in which they themselves are defined as Role Assignment Manager.

All groups that were imported from LDAP servers are scheduled to be updated automatically. For detailed information on changing or deactivating this schedule, see [Scheduling Automatic LDAP Group Synchronization](#).

Scheduling Automatic LDAP Group Synchronization

If you have imported user groups from LDAP servers in your Silk Central system, you can specify a schedule to automatically synchronize your Silk Central groups with changes that were made on the LDAP servers.

To define a schedule for synchronizing Silk Central groups with changes that were made on the LDAP servers:

1. On the **Instance Administration** page, stop the application server of the instance that you want to modify.

2. Open the `TAppServerHomeConf.xml` file with a text editor.

The default path for this file is `C:\Program Files (x86)\Silk\Silk Central 20.6\instance_<instance number>_<instance name>\Conf\AppServer` on the application server.

3. Locate the `LdapUpdate` XML tag.

You can modify the following settings:

ScheduledTime Timestamp (hh:mm, 24h clock) when the data from LDAP is updated the first time. `ScheduledTime` is interpreted with the timezone of the application server system environment. Leave this setting empty to disable automatic synchronization.

UpdateIntervallInMinutes This interval (in minutes) specifies how often LDAP groups and users are synchronized, starting relatively from `ScheduledTime`. Minimum value is 1.

4. Save and close the XML file.
5. Restart the application server.

Editing Groups

To edit a group:

1. In the menu, click **Administration > User Management**.
2. Click the **Groups** tab.
3. Click the group name of the group you want to edit. The **Configure existing user group** page displays.
4. In the **Group Name** field, edit the name as required. This field cannot be edited for groups that were imported from an LDAP server.
5. In the **Description** field, edit the group's description as required.
6. Select a user with a role assignment from the respective list boxes, then click **Add Selection** to add the user and role combination to the new group.



Note: Any user roles that have been defined as not accepting new user assignments are not displayed in this list. These settings are controlled through the **Allow New Assignment** buttons at **Administration > User Management > Roles**.

For groups that were imported from an LDAP server, additional role assignments can be added to the users and role assignments can also be deleted, but at least one role assignment per user must remain. Removing users from an LDAP-imported group can only be done on the LDAP server itself.

7. Repeat the previous step to assign all desired user and role combinations to the group.



Tip: It may be useful to assign the role **Role Assignment Manager** to one or more users of the group. Role Assignment Managers have the sole permission to assign/unassign roles to/from existing users of a group, and only roles that they themselves are assigned to. They cannot add or remove members from a group, and they can only change role assignments within groups in which they themselves are defined as Role Assignment Manager.

8. To remove a user and role combination from the current group, click **X** in the **Actions** column.
9. For groups that were imported from an LDAP server, click **Synchronize with LDAP** to synchronize changes on the LDAP server with Silk Central. If a user has been removed from the group in LDAP, then all their role assignments are removed from the group. If a user has been added in LDAP, a new role assignment with the default role is created. All other role assignments remain as they were.



Note: Group assignments are automatically synchronized when a user logs in to Silk Central.

10. In the **Project Assignment(s)** section you can assign any existing projects to this group.
11. Click **Save** to return to the **Groups** page.

Deleting Groups

Before you can delete a group, you must remove all user and role assignments from the group. This restriction does not apply to groups that were imported from an LDAP server. For additional information about modifying groups, see *Editing Groups*.

To delete a group:

1. In the menu, click **Administration > User Management**.
2. Click the **Groups** tab.
3. In the **Actions** column of the group you want to remove, click **X**. A confirmation dialog box displays.
4. Click **Yes** to confirm the operation; click **No** to abort.

 **Note:** When deleting a group that was imported from an LDAP server, all user/role assignments are also deleted. If a user of the deleted group has no assignments in any other groups, the user will be deleted in Silk Central.

Group Settings Page

Administration > User Management > Groups > New/Edit Group

Use the **Group Settings** page to configure user groups. Group settings are closely related to user account settings. The page displays the following items:

| Item | Description |
|---------------------------------------|--|
| Group name | Specifies the name of the group as it should display in the GUI. You can define any name for the group. |
| Description | A description of the group. You can enter any text for the description. |
| Account and Role Assignment(s) | Lists all existing user/role assignments of the group. You can also delete user and role assignments by clicking X next to the assignment you want to remove. |
| User | This list box lists the user accounts that have been defined by an administrator. Select a user to be assigned to the group. |
| Role Definition | The list is populated with the predefined system roles and the custom user roles. Select the user role with which the user is to be assigned to the group. |
| Add Selection | Click to create a new user account and user role assignment with the selected user and user role. |
| Project Assignment(s) | Lists all existing projects and whether they are assigned to the group account. Check the check box next to a project to assign the project to the group account. If no projects exist, you may assign them later after you have created them. |
| Select All | Checks the check boxes of all listed projects. |
| Deselect All | Un-checks the check boxes of all listed projects. |

Execution Environment

Manage the execution environment to specify where Silk Central tests are executed. Silk Central's execution environment consists of physical and virtual (cloud-hosted) execution servers, which are grouped into logical containers called **Locations**.

Locations are logical containers for execution servers. For information on setting up execution servers, see *Setting Up Execution Servers*. Since Silk Central supports worldwide distribution of Points of Presence (PoP) — the distribution of execution servers — it is desirable to group execution servers into locations.

Locations are not required to be physical locations though, they can simply be used to group your execution servers into manageable units.

Silk Central execution servers are responsible for executing scheduled tests, for example Silk Test Classic and Silk Performer scripts. Execution servers can be either physical or virtual (cloud-hosted) machines.

Managing Locations

Locations are logical containers for execution servers. For information on setting up execution servers, see *Setting Up Execution Servers*. Since Silk Central supports worldwide distribution of Points of Presence (PoP) — the distribution of execution servers — it is desirable to group execution servers into locations. Locations are not required to be physical locations though, they can simply be used to group your execution servers into manageable units.



Note: Silk Central automatically creates a default location called `Local`.

Adding Locations

To add a new location:

1. In the menu, click **Administration > Execution Environment**.
2. Click **New Location**.

The **New Location** dialog box displays.

3. Type a **Location Name**.
4. If you have specified the location of a proxy server, select **Use System Proxy** by checking the respective check box.

For more information, see *Configuring a System Proxy*.

5. In the **Location Proxy** section, you can define a proxy server through which the execution servers of this location will communicate with the application server.



Note: To connect execution servers through a proxy, the execution server secure port must be changed from 19125 to 443. For command-line-driven execution servers, you can specify the port as a parameter. For other execution servers, specify the port in the file `SccExecServerBootConf.xml`. This .xml file is located on the execution server, in the Silk Central directory, in `/conf/execserver`. Make sure that SSL is enabled under **Administration > Execution Environment > Execution Servers**.

6. In the **Host** field, type the name of the computer hosting the proxy service.
7. In the **Port** field, type the port number of the proxy host.
8. If the proxy server requires a username/password authentication, type the valid credentials in the **Username** and **Password** fields.
9. The **Projects** section includes a list of existing projects. Check the **Assigned** check boxes of the projects that you want to assign to this location.
10. Click **OK** to add the new location.

Editing Locations

Describes how to edit a location.

To edit a location:

1. In the menu, click **Administration > Execution Environment**.
2. Select the location that you want to modify and click .

The **Edit Location** dialog box displays.

3. Modify the **Location Name** as required.
4. If you have specified the location of a proxy server, select **Use System Proxy** by checking the respective check box.

For more information, see *Configuring a System Proxy*.

5. In the **Location Proxy** section, you can define a proxy server through which the execution servers of this location will communicate with the application server.



Note: To connect execution servers through a proxy, the execution server secure port must be changed from 19125 to 443. For command-line-driven execution servers, you can specify the port as a parameter. For other execution servers, specify the port in the file `SccExecServerBootConf.xml`. This .xml file is located on the execution server, in the Silk Central directory, in `/conf/execserver`. Make sure that SSL is enabled under **Administration > Execution Environment > Execution Servers**.

6. In the **Host** field, type the name of the computer hosting the proxy service.
7. In the **Port** field, type the port number of the proxy host.
8. If the proxy server requires a username/password authentication, type the valid credentials in the **Username** and **Password** fields.
9. The **Projects** section includes a list of existing projects. Check the **Assigned** check boxes of the projects that you want to assign to this location.
10. Click **OK**.

Deleting Locations



Tip: Before you can delete a location, you must first remove all assigned execution servers from the location. For more information, see *Deleting Execution Servers*.

To delete a location:

1. In the menu, click **Administration > Execution Environment**.
2. Select the location that you want to remove and click **X**. A confirmation dialog box displays, asking you to confirm the deletion.
3. Click **Yes** if you want to remove the location, or click **No** to abort the operation.

New Location Dialog Box

Administration > Execution Environment > Execution Servers > New Location

Use the **New Location** dialog box to configure locations.

| Item | Description |
|-------------------------|--|
| Name | Specifies the name of the location as it should appear in the GUI and in reports. |
| Use system proxy | Enabling this setting will force all execution servers of this location to communicate with the application server through the defined system proxy. If this setting is not enabled, the application server will communicate directly with the execution servers, unless you define a location proxy. This check box is disabled if no system proxy is defined. |
| Location proxy | Use this area to define a proxy server through which the execution servers of this location will communicate with the application server. Leave the fields empty if you want the execution servers of this location to communicate directly with the application server, or if you checked the Use system proxy option. You can also define a system proxy and a location proxy, in which case the communication will be tunneled through both proxies. You may only define a location proxy that supports Secure Sockets Layer (SSL). All execution servers must use the SSL port of the proxy. For detailed information about execution server settings, see <i>Setting Up Execution Servers</i> . |

| Item | Description |
|--------------------------|---|
| Hostname | The name of the computer hosting the proxy service. |
| Port | The port number of the proxy host. Default is port 443. |
| Username | If the proxy server requires a username/password authentication, enter a valid username. |
| Password | If the proxy server requires a username/password authentication, enter a valid password for the user specified in the Username field. |
| Assigned Projects | Lists all existing projects. Check the check box next to a project to assign the project to the location. If no projects exist, you can assign them later after you have created them. For more information, see <i>Adding Projects</i> . Selected projects will have access to the execution servers at this location. |
| Select All | Checks the check boxes of all listed projects. |
| Deselect All | Un-checks the check boxes of all listed projects. |

Setting Up Execution Servers

Silk Central execution servers are responsible for executing scheduled tests, for example Silk Test Classic and Silk Performer scripts. Execution servers can be either physical or virtual (cloud-hosted) machines.

To execute Silk Test Classic and Silk Performer tests, Silk Test Classic or Silk Performer software must be installed on the same computer on which Silk Central execution servers are installed.

When executing Silk Performer scripts against multibyte applications or Web pages, review the *Multibyte Support* section in the *Silk Performer Help*.

For further details regarding the installation of execution servers, see *Installing Silk Central*. For information regarding Silk Test Classic and Silk Performer, refer to the respective product documentation.

Hardware Provisioning with Keywords

The hardware-provisioning technology of Silk Central helps you manage test environments that include numerous execution servers. Rather than having to configure a one-to-one direct relationship between each execution server and execution plan, keywords enable Silk Central to select the most appropriate execution server for each execution plan. This is done through dynamic comparison of an execution plan's keyword list with the keyword lists of all active execution servers.

Keywords typically describe the environment of an execution server, for example platform, operating system, and pre-installed applications. When an execution plan is executed, Silk Central compares the execution plan's keywords with the keywords of all available execution servers. The execution is then run on the execution server that matches the execution-plan's keyword list. If there is no matching execution server, the execution will not be run. If there are multiple execution servers with matching keyword lists, the execution will be run on the first identified execution server.

For physical execution servers on which Silk Test is installed, Silk Central automatically assigns keywords depending on the installed browsers and connected mobile devices.

Reserved Default Keywords

If you do not require hardware provisioning, you can rely on the reserved keywords that are created automatically for each execution server. In such cases, it is not necessary that you manually assign keywords to your execution servers. Instead, you can configure a one-to-one static execution-server assignment for each execution plan.

A reserved keyword is assigned automatically to each newly created execution server. Reserved keywords are structured in the following form:

```
#<execution server name>@<location name>
```

Reserved keywords are available when assigning keywords to execution plans. They are neither available nor applicable when assigning keywords to execution servers.

In addition to the reserved keywords that are set up automatically for each defined execution server, reserved keywords are also set up for each execution server type:

- #PHYSICAL** Limits execution-server provisioning to physical execution servers.
- #VIRTUAL** Limits execution-server provisioning to virtual execution servers.

Keywords and Virtual Execution Servers

Keywords are assigned to virtual execution servers in the same way that they are assigned to physical execution servers. When you configure at least one virtual execution server, the **#VIRTUAL** keyword is dynamically created and made available for assignment to all execution plans. If you prefer that an execution is executed on a virtual machine, select the **#VIRTUAL** keyword for the execution plan. When an execution plan has neither (or both) the **#VIRTUAL** and **#PHYSICAL** keywords, the execution may occur on either a virtual or a physical execution server, assuming the settings of the execution environments are the same. When an execution-plan's keywords match multiple virtual execution servers, the first matching virtual execution server that is identified is selected.

Configuring Physical Execution Servers

Physical execution servers run on a physical computer or on a virtual machine (VM). To configure a physical execution server:

1. In the menu, click **Administration > Execution Environment**.
2. Select the location for which you want to configure an execution server.
 - To create a new execution server, click **New Physical Execution Server**.
 - To edit an existing execution server, click the respective **Edit** button in the **Actions** column.
-  **Note:** Execution servers must be deactivated before their properties can be edited. Keyword-lists of active execution servers can however be edited.
3. Enter a valid IP address or hostname in the **Hostname or IP address** field.
4. Enter a name for the execution server in the **Name** field.
5. Click **Keywords** to select keywords from a list or add new keywords that describe the execution server. These keywords are analyzed at execution time to dynamically select an appropriate server for each execution. For more information, see [Creating and Assigning Execution Server Keywords](#).
6. Settings in the **Advanced** section only need to be modified if you are not using the default settings. Specify the port on which the execution server listens in the **Port** field.
 -  **Note:** Check the **Use SSL** check box if you want to connect to the execution server through SSL.
 -  **Tip:** To connect to the execution server through a non-standard SSL port, see [Configuring a Non-Standard SSL Port for Execution Servers](#).
7. *Optional:* Type a responsiveness timeout in seconds in the **Responsiveness Timeout** field. If the execution server is not accessible for the specified time, an email is sent to the Silk Central administrators. Enter 0 if you do not want Silk Central to send emails in such a case.
8. *Optional:* Change the amount of **Maximum concurrent executions** on this execution server, if required. If you enable parallel execution for your execution plans, this is the maximum amount of execution plans that are allowed to run concurrently on this execution server. Note that allowing too many concurrent executions can have a severe impact on the execution server's performance and may cause test executions to fail.
9. Click **OK** to save your settings. If you receive an error message, ensure that your settings are correct, the network is configured properly, and that the required software is installed on the execution server you are setting up.



Important: Because the installation of an execution server requires administrative privileges, the automatic upgrade of an execution server fails if UAC is enabled. Disable UAC on all computers that host an execution server.



Note: If the version of the execution server is an invalid older version, but later than version *SilkCentral Test Manager 2009 SP1*, the execution server is automatically upgraded to the current Silk Central version. Silk Central shows a message concerning the upgrade in the **Information** column in the list of execution servers. As long as the upgrade procedure is not complete, the upgrading execution server is not used.

Configuring Cloud-Hosted Execution Servers

Cloud-hosted execution servers run on a cloud provider, like VMware vCloud. To configure a cloud-hosted execution server:

1. In the menu, click **Administration > Execution Environment**.

2. Select the location for which you want to configure an execution server.

- To create a new execution server, click **New Virtual Execution Server**.
- To edit an existing execution server, click the respective **Edit** button in the **Actions** column.
- To synchronize the list of execution servers that use templates from a cloud provider, select the

Cloud Providers tab. Click  in the **Actions** column to synchronize the list of execution servers that use templates from this cloud provider. This will remove all virtual execution servers of this profile that have a template assigned which no longer exists. After a Hotfix has been applied, this will also update the templates of automatically created virtual execution servers.



Note: Execution servers must be deactivated before their properties can be edited. Keyword-lists of active execution servers can however be edited.

3. In the **Cloud Provider** list, choose from the list of configured cloud provider profiles to determine in which cloud the virtual environment will be deployed when this execution server is used for an automated run.

4. In the **Virtual Image Template** list, select the image template from which a new instance will be created for each automated run. Depending on the cloud provider, an image template consists of at least one virtual machine image. A Silk Central execution server must be running in one virtual machine image. If the execution server version is outdated, the server will be upgraded automatically after deploying the virtual environment for each automated run. The deployment procedure is quicker if the VM image already runs a current execution server version.

5. Enter a name for the execution server in the **Name** field.

6. Click **Keywords** to select keywords from a list or add new keywords that describe the execution server.

These keywords are analyzed at execution time to dynamically select an appropriate server for each execution. For more information, see [Creating and Assigning Execution Server Keywords](#).

7. Settings in the **Advanced** section only need to be modified if you are not using the default settings. Specify the port on which the execution server listens in the **Port** field.



Note: Check the **Use SSL** check box if you want to connect to the execution server through SSL.



Tip: To connect to the execution server through a non-standard SSL port, see [Configuring a Non-Standard SSL Port for Execution Servers](#).

8. *Optional:* Type a responsiveness timeout in seconds in the **Responsiveness Timeout** field. If the execution server on the cloud provider is not accessible for the specified time, the VM is shut down.

Enter 0 if you do not want to shut down the VM in such a case (this will require you to manually abort the test run if the VM can never be reached).

9. Click **OK** to save your settings and enable the execution server for automated runs according to the specified keywords. If you receive an error message, ensure that your port setting is correct and that an execution server is installed on the virtual machine image. If the error message mentions that the

installed execution server is of an invalid version, you can ignore the error because the server will be upgraded automatically for each automated run.

Creating and Assigning Execution Server Keywords

To create and assign execution server keywords:

1. In the menu, click **Administration > Execution Environment**.
2. Select a location to access the list of defined execution servers for that location.
3. In the **Actions** column of a predefined execution server, click , or click **New Physical Execution Server** or **New Virtual Execution Server** to create keywords for a new execution server.

 **Note:** For physical execution servers on which Silk Test is installed, Silk Central automatically assigns keywords depending on the installed browsers and connected mobile devices. Click  in the **Actions** column to refresh the list of assigned keywords.

4. Click **Keywords**. The **Assign Keywords** dialog box displays.
5. Type a new keyword into the **Select or enter keywords** field or select an existing keyword that describes the environment on the execution server, like the platform, operating system, or pre-installed applications. The following characters cannot be used in keywords: # \$? * \ , ; ' "

 **Note:** If you are working with a new Silk Central installation you may not see any available keywords. Though not visible on the **Assign Keywords** dialog box, reserved keywords are created for each execution server that is configured for the system. These reserved keywords are only available when assigning keywords to execution plans.

You can use **Ctrl+Click** or **Shift+Click** to select multiple keywords using standard Windows multi-select functions. Note that the **Select or enter keywords** field is disabled when multiple keywords are selected in the lists.

If you do not require hardware provisioning, you can use the default, reserved keywords that are created for each execution server. In such cases, it is not necessary that you assign additional keywords to the execution server.

6. Click **Add (>)** to move the keyword into the **Assigned keywords** list.
7. Click **OK** to save the keywords and close the **Assign Keywords** dialog box.

Activating or Deactivating Execution Servers

1. In the menu, click **Administration > Execution Environment**.
2. Select a location to access the list of defined execution servers for that location.
3. In the **Status** column of the execution server you want to activate or deactivate, click **Inactive/Active**.

 **Important:** Because the installation of an execution server requires administrative privileges, the automatic upgrade of an execution server fails if UAC is enabled. Disable UAC on all computers that host an execution server.

 **Note:** If the version of the execution server is an invalid older version, but later than version *SilkCentral Test Manager 2009 SP1*, the execution server is automatically upgraded to the current Silk Central version. Silk Central shows a message concerning the upgrade in the **Information** column in the list of execution servers. As long as the upgrade procedure is not complete, the upgrading execution server is not used.

Deleting Execution Servers

 **Tip:** To prevent data inconsistency, you need to deactivate an execution server before you can delete it. For additional information, see *Activating or Deactivating Execution Servers*.

 **Note:** Deleting an execution server does not remove the actual software installation. Deletion simply disconnects the execution server. You can add a previously deleted execution server again. To assign an execution server to a different application server, you need to delete the following file on the computer where the execution server is installed: `C:\ProgramData\SilkCentral\AgentBase\ItemObjects.ser`.

1. In the menu, click **Administration > Execution Environment**.
2. Select a location to access the list of defined execution servers for that location.
3. In the **Actions** column of the execution server you want to remove, click .

Configuring a Non-Standard SSL Port for Execution Servers

The default SSL port through which the application server communicates with execution servers is 19125.

 **Note:** This procedure needs to be performed for each execution server that you want to connect to through a non-standard SSL port.

To configure a non-standard SSL port for an execution server:

1. Deactivate the execution server for which you want to configure a non-standard SSL port.
2. Stop the execution server.
3. Open the `SccExecServerBootConf.xml` file with a text editor.
This file is located in the `/conf/execserver` folder of the Silk Central directory on the execution server.
4. Locate the `<SSLPort>` XML tag. By default, the tag is set to `<19125>`.
Set the value to the port number that you want to use for SSL communication.
5. Save and close the XML file.
6. In Silk Central, set the SSL port of the execution server to the value that you have specified in the XML file.
7. Restart the execution server.
8. Reactivate the execution server.

New/Edit Execution Server Dialog Box

Administration > Execution Environment > Execution Servers > New/Edit Execution Server

Use the **New/Edit Execution Server** dialog box to configure execution servers within a location.

| Item | Description |
|----------------------------------|---|
| Name | Defines a name for the execution server. This name will appear in all tables and result reports for executions from this specific computer. You can enter up to 100 characters. |
| Description | A description of the execution server. You can enter any text for the description. |
| Physical execution server | Select this option to specify a physical machine as execution server. |
| Host or IP-address | Specifies the name of the host or the IP-address of the computer on which the execution server is installed. Some networks may only find the execution server if you specify the full name of the host, including the name of the domain, for example <code>MyHost.MyDomain</code> . |

| Item | Description |
|-----------------------------------|--|
| Port | Specifies the port of the computer defined in the Host or IP-address field on which the execution server listens. The default port is 19124. |
| Use SSL | <p>Check this check box if you want the application server to connect to the execution server through Secure Sockets Layer (SSL). The default SSL port is 19125.</p> <p>If you selected to use a proxy server for the location to which this execution server is assigned, you must check Use SSL with port 443.</p> |
| Responsiveness timeout [s] | <p>For virtual execution servers: Type a responsiveness timeout in seconds in the Responsiveness Timeout field. If the execution server on the cloud provider is not accessible for the specified time, the VM is shut down. Enter 0 if you do not want to shut down the VM in such a case (this will require you to manually abort the test run if the VM can never be reached).</p> <p>For physical execution servers: Type a responsiveness timeout in seconds in the Responsiveness Timeout field. If the execution server is not accessible for the specified time, an email is sent to the Silk Central administrators. Enter 0 if you do not want Silk Central to send emails in such a case.</p> |
| Keywords | Lists the keywords that have been defined for this execution server. Keywords enable Silk Central's hardware-provisioning technology to dynamically identify the most appropriate execution server for each test execution. Click Keywords to edit the keywords list for this execution server. |

Cloud Integration

Silk Central enables you to integrate with providers of public or private cloud services by configuring cloud provider profiles. Cloud profiles are based on a plug-in concept, which enables you to write your own plug-in for a specific cloud provider. A cloud provider plug-in deploys a virtual environment before each automated test run.



Note: The cloud API is subject to change in upcoming versions of Silk Central. If you use this API, upgrading to a future version of Silk Central may require you to make updates to your implementation.

Adding Cloud Profiles

Add a cloud profile to integrate with providers of public or private cloud services.

1. In the menu, click **Administration > Execution Environment**.
2. Click the **Cloud Providers** tab.
3. Click **New Profile**. The **New Cloud Provider Profile** dialog box appears.
4. Select the cloud provider that you want to access from the **Type** list of installed plug-ins. *VMware vCloud* and *Amazon Web Services* are supported out-of-the box.
5. In the **Name** field, enter a name for the profile. This name will be displayed in the list of profiles when creating a cloud-hosted execution server.

6. Enter valid credentials to remotely access the cloud system in the **Username** and **Password** fields. For Amazon Web Services, this is the **Access Key** and the **Secret Key**.
7. To configure a VMware vCloud profile (skip this step to configure a different profile):
 - a) Enter the URL with the hostname only and do not append `/cloud/org/<organization name>` as you would when accessing the system with a Web browser. Example: `https://vdirectorweb`.
 - b) **Optional: Version:** Leave the field empty if you are using VMware vCloud Director 5.5 (default). Enter `v5.1` if you are using vCloud Director 5.1.
 - c) **Organization:** Enter the name of the vCloud organization that you want to access, for example `dev_org`.
8. To configure an Amazon Web Services profile (skip this step to configure a different profile):
 - a) **Region:** Select the region in which you want to deploy your instance.
 - b) **Instance Type:** The instance types that Amazon offers. This defines for example how much RAM and CPU power your instance will have. Note that this also has an impact on lease costs. For more information, visit the [Amazon EC2 Instances](#) page.
 - c) **Keep instance alive on connection error:** Select `true` if you want the instance to stay alive when the connection between the application server and the Amazon instance breaks.
9. Click **Test Connection** to perform a test connection to the cloud provider with the information that you have entered.
10. Click **OK** to save your settings.

If the selected cloud provider has templates named "Micro Focus Silk Central Execution Server 20.6.0.x", these templates are automatically imported as virtual execution servers. Only templates with exactly the same version as your running Silk Central environment (see **Help > About**) will be imported.

Editing Cloud Profiles

Edit a cloud profile to modify how you access the provider of a public or private cloud service.

 **Important:** If virtual environments based on this profile are already deployed (see the **Deployed Instances** column of the grid), changing the settings might cause that Silk Central will not delete these unused environments automatically anymore.

1. In the menu, click **Administration > Execution Environment**.
2. Click the **Cloud Providers** tab.
3. Click  in the **Actions** column to synchronize the list of execution servers that use templates from this cloud provider. This will remove all virtual execution servers of this profile that have a template assigned which no longer exists. After a Hotfix has been applied, this will also update the templates of automatically created virtual execution servers.
4. Click  (**Edit**) in the **Actions** column of the profile that you want to edit. The **Edit Cloud Provider Profile** dialog box appears.
5. In the **Name** field, enter a name for the profile. This name will be displayed in the list of profiles when creating a cloud-hosted execution server.
6. Enter valid credentials to remotely access the cloud system in the **Username** and **Password** fields. For Amazon Web Services, this is the **Access Key** and the **Secret Key**.
7. To configure a VMware vCloud profile (skip this step to configure a different profile):
 - a) Enter the URL with the hostname only and do not append `/cloud/org/<organization name>` as you would when accessing the system with a Web browser. Example: `https://vdirectorweb`.
 - b) **Optional: Version:** Leave the field empty if you are using VMware vCloud Director 5.5 (default). Enter `v5.1` if you are using vCloud Director 5.1.
 - c) **Organization:** Enter the name of the vCloud organization that you want to access, for example `dev_org`.

8. To configure an Amazon Web Services profile (skip this step to configure a different profile):
 - a) **Region:** Select the region in which you want to deploy your instance.
 - b) **Instance Type:** The instance types that Amazon offers. This defines for example how much RAM and CPU power your instance will have. Note that this also has an impact on lease costs. For more information, visit the [Amazon EC2 Instances](#) page.
 - c) **Keep instance alive on connection error:** Select `true` if you want the instance to stay alive when the connection between the application server and the Amazon instance breaks.
9. Click **Test Connection** to perform a test connection to the cloud provider with the information that you have entered.
10. Click **OK** to save your settings.

Deleting Cloud Profiles

Delete a cloud profile if you no longer need to access a specific provider of a public or private cloud service.

 **Important:** Deleting a profile will also delete all execution servers that use this profile and abort all executions that currently use these execution servers. If virtual environments based on this profile are already deployed (see the **Deployed Instances** column of the grid), changing the settings might cause that Silk Central will not delete these unused environments automatically anymore.

1. In the menu, click **Administration > Execution Environment**.
2. Click the **Cloud Providers** tab.
3. Click  (**Delete**) in the **Actions** column of the profile that you want to remove. A confirmation dialog box displays.
4. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

VMware vCloud Integration

Mappings

The templates that are available for vCloud execution servers correspond to vApp templates in local (non public) catalogs in vCloud. They are displayed as `<catalog name> - <vApp template name>`.

Deploying a virtual environment for a test run creates a new vApp in vCloud Director, with its name based on the template name and the suffix `- for Silk Central <timestamp>`.

Guest Operating System Customization Requirements

To enable the Silk Central execution server to start up properly, the Guest Operating System Customization must be activated and configured in the vApp template virtual machine properties in vCloud Director as outlined below. As the execution server is started in a user session, a Windows login must only occur once after the VM reboot is initiated by the Guest Operating System Customization.

- Enable **Allow Local Administrator Password** and **Login as Administrator automatically**
- Specify the password or let vCloud generate a new one
- Disable **Administrator must change password on first login**
- Make sure that in Windows itself, no **Auto login** is activated, as this would cause a duplicate login and execution server startup due to the customization reboot

Limitations

- The Silk Central vCloud integration only allows the usage of vApp templates in local (non public) catalogs. Public catalogs can still be used to copy or move templates across organizations, but the contained templates will not be available in Silk Central. The reason for this limitation is that there are potential problems to create vApps from templates in a public catalog if the template originates from

another organization. This happens because the vDC and the network associated with the template are not valid in another organization.

- No fencing support. Silk Central does not support that multiple VMs within a vApp communicate with each other with a known IP address, because each deployed VM dynamically receives a new IP.
- vApp templates with the setting **Create Identical Copy** are not supported. When creating a vApp template to be usable in Silk Central, make sure that in the properties of the vApp template the option **Customize VM settings** is enabled. Setting this option to **Identical Copy** will raise an error when Silk Central attempts to deploy vApps from this template. Note that this option can only be set when creating the vApp template.
- Accessing commercial service providers that expose a vCloud REST API has not been tested, but may work. The integration has been tested with VMware vCloud Director 5.5 installed as private cloud.
- To prevent IP assignment conflicts, store vApp templates with DHCP IP mode configured for the network adapter. If this is not configured, booting a VM for the first time after deployment can cause vApp machines to have an old static IP from the time the template was stored. The correct IP gets updated when VMware does a guest operating system customization, which adjusts the IP configuration and initiates a reboot. Silk Central always configures static IP pool mode (applied after guest operating system customization), no matter what is configured in the template.
- If using more than one network in the virtual data center (vDC), configure the network to be used in the stored vApp template. Silk Central matches the name of the network that is set in the template VM network adapter with the vDC network names to choose the right vDC network when deploying a new vApp.
- If an execution server older than version 15.0 is installed within the template VM, it must run in **As Process** mode

Cloud Providers Page

Administration > Execution Environment > Cloud Providers

The **Cloud Providers** page lists all configured cloud provider profiles. Use this page to manage your cloud provider profiles.

On this page you can perform the following actions:

- Click **New Profile** to configure a new cloud provider profile.
- Click  (**Edit**) in the **Actions** column to edit the settings.
- Click  (**Delete**) in the **Actions** column to delete a cloud provider profile.
- Click  in the **Actions** column to synchronize the list of execution servers that use templates from this cloud provider. This will remove all virtual execution servers of this profile that have a template assigned which no longer exists. After a Hotfix has been applied, this will also update the templates of automatically created virtual execution servers.

Managing Device Providers

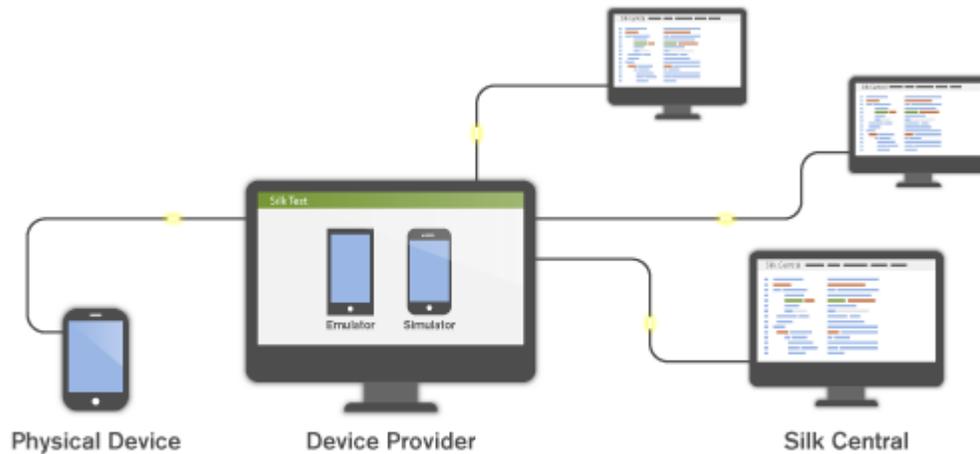
Device providers are computers that host mobile emulators and simulators, and/or have physical mobile devices connected to them, which can be accessed to execute tests.

Physical mobile devices, emulators and simulators that are connected to or installed on a so called device provider, can be made available to your test engineers and test plans. Manual and automated tests can then be executed directly on those mobile devices.

Today's software usually needs to run on multiple platforms and on all kinds of mobile devices, so this feature is especially useful to perform configuration testing across all platforms and devices that your software is supporting.

Configuring Device Providers

To test your applications on mobile devices (physical devices, emulators and simulators), configure a device provider.



Before configuring specific device providers, ensure that the installation prerequisites are met:

- **Silk Test Information Service:** Device providers require that you first install the Silk Test Information Service on the computer(s) that will host your mobile devices.
 1. Download the setup file from [Micro Focus SupportLine](#).
 2. Execute the setup file and simply follow the instructions on the Installation Wizard. For information about setting up a device provider on a Mac, see [Installing the Silk Test Information Service on a Mac](#).
- **UFT Mobile and Sauce Labs:** Refer to the Silk Test documentation:
 - [Testing mobile applications with Silk Test Workbench](#)
 - [Testing mobile applications with Silk4J](#)
 - [Testing mobile applications with Silk4NET](#)
 - [Testing mobile applications with Silk Test Classic](#)

To configure a device provider:

1. In the menu, click **Administration > Execution Environment**.
2. Click the **Device Providers** tab.
3. Select the location for which you want to configure a device provider.
 - To create a new device provider, click **New Device Provider**.
 - To edit an existing device provider, click the respective **Edit** button in the **Actions** column.

 **Note:** The same device provider can be added to multiple locations.

4. Select the **Device Provider Type**.
 - **Silk Test Information Service:** Enter the Silk Test **Information Service URL** of the computer that hosts the mobile devices. This can be an IP address, a URL, or a hostname, for example `http://<computer name>:22901`. If no port is specified, the default port 22901 is appended.
 - **Sauce Labs:** Enter your **Sauce Labs user** name and your Sauce Labs **Access Key**. To find your access key, login to <https://saucelabs.com> and select **My Account**.



Note: If you plan to use Sauce Labs devices for manual testing, a **Silk Test Information Service** must be selected from the list. If no Silk Test Information Service is selected, Sauce Labs devices will only be available for automated testing.

- **UFT Mobile:** Enter your UFT Mobile **URL** and your **User** name and **Password** for automated testing. For manual testing, you will have to enter the credentials when the UFT Mobile web UI opens.

5. In the **Name** field, enter a speaking name for the device provider.

6. Click **Test Connection**. If the test connection was successful, click **OK**.

Installing the Silk Test Information Service on a Mac



Note: To install the information service on a Mac, you require administrative privileges on the Mac.

To create and execute tests on a Mac using Apple Safari or using iOS or Android devices, install the Silk Test information service (information service) on the Mac. Once the information service is installed and active, you can record and replay tests from a Silk Test client that is installed on a Windows machine.



Note: You cannot record on a Mac. To add a Mac as a test location to Silk Test, see *Editing Remote Locations* in the Silk Test documentation. .

To install the information service on a Mac:

1. Read the information in the topic *Prerequisites for Testing with Apple Safari on a Mac* in the Silk Test documentation.

2. Ensure that a Java JDK is installed on the Mac.

3. If you want to test mobile applications on an iOS device, ensure that Xcode is installed on the Mac.

4. Access the information service setup file, `SilkTestInformationService<Version>-<Build Number>.pkg`.

- If you have downloaded the information service setup file while installing Silk Test, open the folder `macOS` in the Silk Test installation directory, for example `C:\Program Files (x86)\Silk\SilkTest`.
- If you have not downloaded the information service setup file while installing Silk Test, you can download the setup file from [Micro Focus SupportLine](#).

5. Copy the file `SilkTestInformationService<Version>-<Build Number>.pkg` to the Mac.

6. Execute `SilkTestInformationService<Version>-<Build Number>.pkg` to install the information service.

7. Follow the instructions in the installation wizard.

8. When asked for the password, provide the password of the currently signed in Mac user.

9. When Apple Safari opens and a message box asks whether to trust the SafariDriver, click **Trust**.



Note: If you want to test against Apple Safari 10 or prior on macOS 10.12 (Sierra) or prior, SafariDriver needs to be installed on the Mac. You can only install the SafariDriver if you are directly logged in to the Mac, and not connected through a remote connection.

10. To complete the installation, the installer logs the current Mac user out. To verify that the information service was installed correctly, log in to the Mac.

11. If you are installing the information service on a Mac with macOS Mojave (10.14) or later, you might have to enable additional automation permissions for Silk Test after logging in to the Mac.

If permissions need to be granted, Silk Test will automatically show request permission dialogs.

a) Click **OK** to acknowledge the information dialog.

b) Click **OK** in all sub-subsequent request permission dialogs.



Important: If you do not enable these permissions for Silk Test, you will not be able to test web applications against Google Chrome or mobile applications on an iOS device or on a Simulator on

this Mac. If by mistake you have clicked **Don't Allow** in one of the permission dialogs, open a terminal on the Mac and type the following command:

```
sudo tccutil reset AppleEvents
```

Then restart the Mac and accept the permission dialogs by clicking **OK**.

12. Click on the Silk Test icon in the top-right corner of the screen to see the available devices and browsers.



Tip: If the Silk Test icon does not appear, restart the Mac.

Deleting Device Providers

Delete a device provider if the connected devices should no longer be used for test executions.

1. In the menu, click **Administration > Execution Environment**.
2. Select a location to access the list of defined device providers for that location.
3. In the **Actions** column of the device provider you want to remove, click **X**.

Managing Report Templates

Silk Central offers a variety of pre-installed reports that let you quickly and easily transform data into presentation-quality information for analysis. The default reports can be customized with either Microsoft Excel, Microsoft Word, or BIRT, an Eclipse-based, open source reporting tool for Web applications. You can also use these tools to create entirely new reports.

Because shared global reports are available in all projects for all users, user roles that are able to update and delete them require special permissions: **Manage shared global reports** and **Delete shared global reports**. These permissions also apply to report templates that are used in shared global reports. Non-shared global reports and report templates can be updated and deleted by user roles with the standard report permissions for updating and deleting reports.

Silk Central reports do not support bitmap (.bmp) image file format. For proper display, images must be in JPEG, GIF, or PNG format.

Managing Custom Report Templates with BIRT

Silk Central is tightly integrated with Business Intelligence and Reporting Tools (BIRT) Report Designer to make it easy for you to generate reports for your test-, build-, defect-, and requirement data.

After downloading a copy of BIRT Report Designer, you can customize the core Silk Central reports and add your own reports. For information about running and customizing reports, please refer to the application's Help.

For additional information on BIRT Report Designer, refer to BIRT Report Designer's online help system. You can find further information, examples, and demonstrations for BIRT Report Designer at <http://www.eclipse.org/birt>. An active newsgroup (news.eclipse.org) is also available.

The software prerequisites to work with BIRT custom reports are:

- BIRT Report Designer
- Access to Silk Central with administrator privileges



Note: Silk Central reports do not support bitmap (.bmp) image file format. For proper display, images must be in JPEG, GIF, or PNG format.

Installing BIRT from Silk Central

This procedure explains how to install BIRT Report Designer from your Silk Central installation. By installing BIRT this way, all necessary configurations for Silk Central are done automatically.

To install BIRT from Silk Central:

1. Ensure that your system uses a 64bit Java Development Kit (JDK) or Java Runtime Environment (JRE). The BIRT Report Designer will not work on a system that uses a 32bit JDK or JRE.
2. Navigate to **Help > Tools** .
3. Click the **BIRT RCP Report Designer** link.
4. After downloading the compressed installer package to your local system, extract the compressed files to a directory on your system, for example `C:\BIRT`.



Note: If you encounter an error when extracting the installer files using Windows compressed folder functionality, use an extraction tool instead, for example WinZip or WinRAR, to extract the files.

5. Start `BIRT.exe` from the directory you extracted the files to.

Establishing Database Access For a New Report Template

Before you can create a new report template with BIRT Report Designer, you need to establish database access to the Silk Central repository you want to query.

To establish database access for a new report template:

1. From within BIRT Report Designer, select the menu **File > New > New Report** .
2. Follow the steps in the **New Report** wizard.
3. Open the **Resource Explorer**.
4. In the **Resource Explorer**, click **Shared Resources > conf > birt > library.rptlibrary > Data Sources > Data Source** and drag the required data source into your report's `Data Sources` directory, which is located in the **Outline** window.
5. In the **Resource Explorer**, click **Shared Resources > conf > birt > library.rptlibrary > Report Parameters** and drag the four report parameters `sourceUser`, `sourcePassword`, `sourceURL`, and `sourceDriver` into your report's `Report Parameters` directory, which is located in the **Outline** window.
6. Double-click the newly imported data source to open the **Edit Data Source** dialog box.
7. Type a valid **Driver Class** and **Database URL**.
For additional information, see **BIRT Data Source Settings** topic.
8. Click **Test Connection** to test your settings. If the database connection has been established, you can proceed with designing your new report template.
9. Click **OK**.

BIRT Data Source Settings

Use the BIRT **New JDBC Data Source Profile** dialog box to establish database access to an existing Silk Central repository. To access the **New JDBC Data Source Profile** dialog box, right-click **Data Sources** in the **Outline** pane, click **New Data Source**, select **JDBC Data Source**, and click **Next >**.

To connect to a MS SQL Server database, use the following credentials:

| Item | String |
|---------------------|---|
| Driver Class | <code>net.sourceforge.jtds.jdbc.Driver</code> |
| Driver URL | <p>MS SQL Server (default instance) <code>jdbc:jtds:sqlserver://<HOST>:<PORT>/<DATABASE></code></p> <p>MS SQL Server (named instance) <code>jdbc:jtds:sqlserver://<HOST>:<PORT>/<DATABASE>;instance=<INSTANCENAME></code></p> |

| Item | String |
|---------------------|---|
| HOST | Host name or IP-address of the computer hosting the database server. |
| PORT | Port number of the database management system. Default is 1433. |
| DATABASE | The name of the database. |
| INSTANCENAME | Instance name of the database instance. The default MS SQL Server Express instance is <code>SQLExpress</code> . |

To connect to an Oracle database, use the following credentials:

| Item | String |
|---------------------|---|
| Driver Class | <code>oracle.jdbc.OracleDriver</code> |
| Driver URL | <code>jdbc:oracle:thin:@<HOST>:<PORT>:<DATABASE></code> |
| HOST | Host name or IP-address of the computer hosting the database server. |
| PORT | Port number of the database management system. Default is 1521. |
| DATABASE | Oracle SID. |

Adapting Existing Report Templates

Silk Central allows you to download and adapt BIRT report templates that contain all the information you need to create custom report templates for use with Silk Central modules.



Note: Silk Central reports do not support bitmap (.bmp) image file format. For proper display, images must be in JPEG, GIF, or PNG format.

To create a report based on a Silk Central template:

1. In the menu, click **Administration > Report Templates**. The **Report Templates** page displays, listing all of the report templates that have been uploaded.
2. Click  in the **Actions** column.
3. Save the template file `<filename>.rptdesign` to the workspace folder of your BIRT installation.
4. Open the downloaded template file in BIRT Report Designer.
5. Redesign the report as necessary.

For instructions on report design, refer to BIRT Report Designer's online help system.

6. To preview your report, choose **View Report > As HTML** from the **Run** menu.

The browser in which you want to preview the report can be specified as follows:

- Click **Window > Preferences > Web Browser**, select **Use external web browser** and choose a browser.
- Click **Window > Preferences > Report Design > Preview** and check the **Always use external browsers** check box.

7. If you preview the report for the first time, the **Enter Parameters** dialog box opens, where you need to specify a valid session ID. To generate a session ID, execute the following URL in a web browser.

```
http://<HOST>:<PORT>/services/sccsystem?
method=logonUser&userName=<USERNAME>&plainPasswd=<PASSWORD>.
```

| Parameter | Description |
|-------------|---|
| HOST | Host name or IP-address of the computer hosting Silk Central. |

| Parameter | Description |
|-------------------------------|---|
| PORT | Port number of the Silk Central front-end server. Default is 19120 if you access Silk Central through a standalone Web server, and 80 if you access Silk Central through IIS. |
| USERNAME/ PASSWORD | Valid credentials of a Silk Central user. |



Note: The order of the valid credentials USERNAME and PASSWORD is very important.

- If at some point your edited report does not return any data, the likely cause is that the session ID has timed out (timeout is 10 minutes). Close the browser window and choose **View Report > As HTML** from the **Run** menu again. To generate a new session ID, repeat the previous step.

Editing Report Template Properties

Once you have created a new custom report template using BIRT Report Designer, Word or Excel and uploaded it to Silk Central, you can edit the template's properties like its name, description, or for which modules the template can be used.

- In the menu, click **Administration > Report Templates** . The **Report Templates** page displays, listing all of the report templates that have been uploaded.
- Click the name of the report template for which you would like to edit or set permissions and associations. The **Edit Report Template** dialog box displays.
- Use the **Name**, **File name** and **Description** fields to edit the template properties.
- In the **Modules** list, you can select if the report template is available for **Issue Manager** and/or for **Test Manager** reports.
- Once you are done editing, click **OK** to save your changes to the report template.
The edits you have made are applied immediately. Users will see changes the next time they access or refresh the report list.

Downloading Report Templates

The report template of the selected report, including the layout, is downloaded. Downloading Silk Central report templates to your local system enables you to edit them. After you download and edit a report, you can upload it to make it available to other users. For more information, see *Uploading Report Templates*.

To download a Silk Central report template:

- In the menu, click **Administration > Report Templates** . The **Report Templates** page displays, listing all of the report templates that have been uploaded.
- Click in the **Action** column of the report you want to download. The **File Download** dialog box displays.
- Click **Save** and download the report file to your local system, depending on the report type that you are downloading.
- Now edit the report based on your needs using either BIRT Report Designer for `rptdesign` files, Word for `docx` files, or Excel for `xls/xlsx/xlsm` files.

Uploading Report Templates

Uploading Silk Central report templates makes them available for others to use. You may want to upload a report template after you have edited it with BIRT Report Designer, Microsoft Word, or Microsoft Excel. You can only run a report if you have access to the project and module to which the report is associated.

 **Tip:** Templates must be configured with additional information so that they can be identified once they are uploaded to Silk Central.

 **Note:** Silk Central reports do not support bitmap (.bmp) image file format. For proper display, images must be in JPEG, GIF, or PNG format.

To upload a customized template as a new report:

1. In the menu, click **Administration > Report Templates** . The **Report Templates** page displays, listing all of the report templates that have been uploaded.
2. Click **Upload** at the bottom of the page. The **Upload Report Template** dialog box displays.
3. Type a **Name** for the report.
4. *Optional:* Type a **Description** of the report.
5. From the **Modules** list box, select the modules with which the report is to be associated.
Hold down the **Ctrl** key to select multiple modules.
6. Click **Browse** next to the **File** field.
7. Browse to and select the template file that is to serve as the basis for the report template.
The file you select must have the `rptdesign`, `docx`, `xlsx`, `xlsm`, or `xls` file extension.
8. Click **OK** to upload the report template for use in Silk Central.

Updating Report Sources

Updating an existing Silk Central report template allows you to move a report you have customized with BIRT Report Designer, Microsoft Word, or Microsoft Excel into Silk Central and make it available to other users.

 **Caution:** Report templates that ship with Silk Central are automatically patched when you upgrade to a new version. It is therefore important that you save your customized report templates in a dedicated custom folder, or that you upload customized report templates as new templates. For more information, see **Uploading Report Templates**.

To update a report template with a modified template file:

1. In the menu, click **Administration > Report Templates** . The **Report Templates** page displays, listing all of the report templates that have been uploaded.
2. Click  in the **Action** column of the report you want to update.
3. Click **Browse** on the **Update Report Template** dialog box to browse to and select the template file that is to overwrite the existing template file.
The file you select must have the `rptdesign`, `docx`, `xlsx`, `xlsm`, or `xls` file extension.
4. Click **OK** to upload the file, and thereby overwrite the file that the report template was previously based on.

Deleting Report Templates

You can remove a Silk Central report from the list of available reports.

To delete a Silk Central report:

1. In the menu, click **Administration > Report Templates** . The **Report Templates** page displays, listing all of the report templates that have been uploaded.
2. Click  in the **Action** column of the report you want to remove. A confirmation dialog box displays.
3. Click **Yes** to remove the report from the list.

Report Templates Page

Administration > Report Templates

Use the **Report Templates** page to manage the report templates which you want to make available to Silk Central for reporting. Report templates are globally available to all reports across all projects.

Click **Upload** to upload a new report template from your hard disk or a UNC to Silk Central.

For each listed report, the page displays the following columns:

| Column | Description |
|--------------------|--|
| Title | The name of the report template as it displays in the application's GUI. |
| File Name | The physical file name of the report template. |
| Uploaded On | Date when the report template was uploaded to Silk Central. |
| Uploaded By | The user who uploaded the report template to Silk Central. |
| Module | The Silk Central application which may access the reporting template. If a template is assigned to no module, then any application can use it. |
| Actions | This column contains action icons which allow the user to perform the following actions on a report template: Update Replaces the currently uploaded template with a new one. Download Downloads the template to your local computer. Delete Deletes the template permanently. |

Audit Log

Administration > Log Files > Audit Log

The audit log allows administrators to view all recorded Silk Central user activity. The log file stores all login and logout information, as well as all changes to the Silk Central database, for example projects, tests, and schedules.

To be able to view audit logs, ensure you have the **View audit logs** permission.

You can manage the listed log entries to suit your information needs by using the available features.

Sorting Data by Column

Clicking a column header sorts all listed data by that column. Clicking the same column header multiple times toggles the sort order between ascending and descending.

Selecting a Range From the Calendar

Click the displayed time range to expand the calendar. The **From** and **To** rows of the calendar allow you to specify start and end times for the period of time for which you want to view data. After specifying **From** and **To** times with the list boxes, click **Update** to update the audit log based on the new time range.

The **day**, **week**, **month**, **quarter**, **[last 7 days]**, **[last 31 days]** links allow you to bypass the calendar and instead view information for set time periods.

You can also use the **Forward** and **Backward** arrows to increase and decrease the selected time range by the following intervals:

- one day
- one week
- one month
- one quarter

Use  and  for increasing and decreasing the range of time covered by the audit log. Clicking  one time enlarges the period of time by 50%. Clicking  one time reduces the period of time by 50%.

When the calendar displays a custom interval, for example after zooming in or out, you can use the left-most arrows, **Earlier** and **Later**, to move the selected period of time forward or backward in time by half of the selected interval.



Tip: After specifying a new time period, click **Update** to update the report.

Filtering Data

Filter options enable you to better target the audit log information you want to analyze.

You can filter listed data by:

- Login** Displays the actions of a specified user login.
- Object** Displays actions taken on a specified database item, for example project or location.
- Operation** Displays selected operations, for example login, logoff, create, or delete.

Accessing and Viewing the Audit Log

To view the audit log:

1. Ensure that you have the **View audit logs** permission.
2. In the menu, click **Administration > Log Files**.
3. Click the **Audit Log** tab.
4. Select a calendar range to limit the listed log entries.
5. Use the filter options to better target the audit log information you want to analyze.

Audit Log Page

Administration > Log Files > Audit Log

Use the **Audit Log** page to view all recorded Silk Central user activity.

To be able to view audit logs, ensure you have the **View audit logs** permission.

| Item | Description |
|---------------|---|
| Calendar area | Select a calendar range to limit the listed log entries. |
| Filter area | Use the filter options to better target the audit log information you want to analyze. Click Update to refresh the list according to your filter settings. |
| Result area | This section displays the logged information. Use the page numbers to move between pages. Click the column headers to sort by the defined column. |

For detailed information about the calendar and filtering options, see *Audit Log*.

Server Log Files

The front-end server, the application server, and the execution server write log files. These files provide valuable information for error analysis. Silk Central allows administrators to view, search, and download these files directly from its Web interface.

Downloading Server Log Files

You can download a server log file to your local computer in CSV format to allow for further data analysis, for example in Microsoft Excel.

To download a server log file:

1. In the menu, click **Administration > Log Files**.
2. Click the tab of the server to which the log file belongs.
 - **Front-end Server Log**
 - **Application Server Log**
 - **Execution Server Log**

A list of log files is displayed in chronological order. Log file names are made up of server component name and a suffix with a timestamp. The current log files are named `FrontendServer.log`, `AppServer.log`, and `ExecServer.log`.



Note: To locate an execution server log file, navigate to the respective execution server through its location.

3. In the **Actions** column of the log file, click .
Alternative: To view the contents of the log file before downloading it, click the name of the log file you want to download. The selected log file displays, along with chronologically sorted log entries. Click **Download as CSV** at the bottom of the page.
4. To view the data in a spreadsheet program, select **Open** on the subsequent dialog box. To save the data on your hard drive, select **Save** on the subsequent dialog box.

Analyzing Server Log Files

To analyze a server log file:

1. In the menu, click **Administration > Log Files**.
2. Click the tab of the server to which the log file belongs.
 - **Front-end Server Log**
 - **Application Server Log**
 - **Execution Server Log**

A list of log files is displayed in chronological order. Log file names are made up of server component name and a suffix with a timestamp. The current log files are named `FrontendServer.log`, `AppServer.log`, and `ExecServer.log`.



Note: To locate an execution server log file, navigate to the respective execution server through its location.

3. Click the name of the log file you want to view. The selected log file is displayed, along with chronologically sorted log entries.
4. Filter options allow you to page recorded log information.

You can filter listed data by:

- Severity** Displays events of a selected severity.
- Log level** Displays events that match a selected log level. More detailed log information can only be displayed when the log level is set accordingly on the server. For more information about configuring a server's log level, see *Log Levels*.
- Module** Displays log information for a selected module. Log entries can only be displayed when the respective products (modules) are installed and connected to the front-end server that is being accessed.

Deleting Server Log Files

 **Caution:** Deleting a log file permanently removes the file from the server. You will not be able to view log data from the deleted file anymore.

To delete a server log file:

1. In the menu, click **Administration > Log Files**.
2. Click the tab of the server to which the log file belongs.
 - **Front-end Server Log**
 - **Application Server Log**
 - **Execution Server Log**

A list of log files is displayed in chronological order. Log file names are made up of server component name and a suffix with a timestamp. The current log files are named `FrontendServer.log`, `AppServer.log`, and `ExecServer.log`.



Note: To locate an execution server log file, navigate to the respective execution server through its location.

3. In the **Actions** column of the log file you want to delete, click **X**. A confirmation dialog box displays.
4. Click **No** to avoid deleting the log file; or click **Yes** to remove the log file from the list.

If you choose **Yes**, the list of log files redisplay, with the deleted log file no longer listed.

Log File Management

Each of the Silk Central servers writes its activities to log files. For more information about Silk Central servers, see *Silk Central Architecture*. When application errors or system failures occur, these log files provide valuable information regarding the root causes of problems. You can customize the level of detail that is written to server log files.

The log files for the Silk Central servers are accessible through **Administration > Log Files**.

Changing Log Levels of the Silk Central Servers

The following servers generate log files:

- Front-end server
- Application server
- Execution server

To change the log level of a Silk Central server:

1. Stop the server for which you want to change the log level.
2. Open the appropriate file with a text editor, depending on the server for which you want to change the log level:

- Front-end server** `SccFrontendBootConf.xml`, located in the `/conf/frontendserver` folder of the Silk Central directory on the front-end server.
- Application server** `SccAppServerBootConf.xml`, located in the `/conf/appserver` folder of the Silk Central directory on the application server.
- Execution server** `SccExecServerBootConf.xml`, located in the `/conf/execserver` folder of the Silk Central directory on the execution server(s).

3. Locate the `<LogLevel>` XML tag in the `<Log>` section of the file.
4. Set the value to the log level at which you want the server to write information. The following log levels are available:

| Value | Log level | Description |
|-------|-----------|---|
| 0 | Overview | The server writes only the most important information to the log files. This is the default setting. |
| 1 | Detailed | The server writes additional information to the log files: <ul style="list-style-type: none"> Front-end server Connection- and event-dispatcher information. Application server Result-writer and result-fetcher activities. Execution server Transaction-execution activities. |
| 2 | Verbose | The server writes additional information to the log files: <ul style="list-style-type: none"> Front-end server User administration information, for example cookie management. Application server Detailed result-writer and result-fetcher information. Execution server Detailed transaction-execution and bandwidth information. |
| 3 | Debug | This is the most detailed log level and should only be used for debugging severe issues. |

5. Save and close the XML file, then restart the server.

Front-End Server Log Page

Administration > Log Files > Front-end Server Log

Use this page to view logging information from the Silk Central front-end server service.

For each log file, the page displays the following columns:

| Column | Description |
|----------------|--|
| Actions | Click the buttons  and  to Delete or Download log files. |
| Name | The name of the log file. |
| Size | The physical size of the log file. |
| Date | Date when the log file was last physically saved. |

Administration > Log Files > Front-end Server Log > Front-end server log file name .

When clicking on the name of a log file, the logging details list displays. The list includes the following items:

| Item | Description | | | | |
|------------------|--|-----------------|---|------------------|--|
| Filter area | Use the filter options to filter the log list information by <i>severity</i> , <i>log level</i> , and <i>module</i> . Click Update to refresh the list according to your filter settings. | | | | |
| Table area | Displays the following logging information: <table border="0" style="margin-left: 20px;"> <tr> <td style="vertical-align: top;">Severity</td> <td>Severity of the event: <ul style="list-style-type: none"> • Info • Warning • Error </td> </tr> <tr> <td style="vertical-align: top;">Log Level</td> <td>Log level of the event: <ul style="list-style-type: none"> • OV = Overview • DT = Detailed • VB = Verbose • DB = Debug </td> </tr> </table> | Severity | Severity of the event: <ul style="list-style-type: none"> • Info • Warning • Error | Log Level | Log level of the event: <ul style="list-style-type: none"> • OV = Overview • DT = Detailed • VB = Verbose • DB = Debug |
| Severity | Severity of the event: <ul style="list-style-type: none"> • Info • Warning • Error | | | | |
| Log Level | Log level of the event: <ul style="list-style-type: none"> • OV = Overview • DT = Detailed • VB = Verbose • DB = Debug | | | | |

Click **Back** to return to the **Front-end Server Log** page. Click **Download as CSV** to download the log file as a CSV file to your local computer.

Application Server Log Page

Administration > Log Files > Application Server Log

Use this page to view logging information from the Silk Central application server service.

For each log file, the page displays the following columns:

| Column | Description |
|----------------|--|
| Actions | Click the buttons  and  to Delete or Download log files. |
| Name | The name of the log file. |
| Size | The physical size of the log file. |
| Date | Date when the log file was last physically saved. |

Administration > Log Files > Application Server Log > Application server log file name .

When clicking on the name of a log file, the logging details list displays. The list includes the following items:

| Item | Description | | | | |
|------------------|--|-----------------|---|------------------|-------------------------|
| Filter area | Use the filter options to filter the log list information by <i>severity</i> , <i>log level</i> , and <i>module</i> . Click Update to refresh the list according to your filter settings. | | | | |
| Table area | Displays the following logging information: <table border="0" style="margin-left: 20px;"> <tr> <td style="vertical-align: top;">Severity</td> <td>Severity of the event: <ul style="list-style-type: none"> • Info • Warning • Error </td> </tr> <tr> <td style="vertical-align: top;">Log Level</td> <td>Log level of the event:</td> </tr> </table> | Severity | Severity of the event: <ul style="list-style-type: none"> • Info • Warning • Error | Log Level | Log level of the event: |
| Severity | Severity of the event: <ul style="list-style-type: none"> • Info • Warning • Error | | | | |
| Log Level | Log level of the event: | | | | |

| Item | Description |
|------|--|
| | <ul style="list-style-type: none"> • OV = Overview • DT = Detailed • VB = Verbose • DB = Debug |

Click **Back** to return to the **Application Server Log** page. Click **Download as CSV** to download the log file as a CSV file to your local computer.

Execution Server Log Page

Administration > Log Files > Execution Server Log

Use this page to view logging information from the Silk Central execution server service.

For each location, the page displays the following columns:

| Column | Description |
|--------------------------|---|
| Location | Displays all available locations. |
| Execution Servers | Displays the amount of execution servers per location. |
| Status | Displays a summary status of the execution servers in the location. |

Administration > Log Files > Execution Server Log > Location name

When clicking on the name of a location, the list of execution servers in the selected location displays. The list displays the following columns for each execution server.

| Column | Description |
|------------------------------|--|
| Execution Server Name | The name of the execution server. |
| Host | The name of the computer hosting the execution server. |
| Type | The Silk Central application that the execution server is configured for. For Silk Central, the type is always <i>Silk Central</i> . |
| Assigned Tasks | The amount of tasks that are currently scheduled on the execution server. |
| Status | The status of the execution server. <i>Active</i> or <i>Inactive</i> . |

Click **Back** to return to the list of locations.

Administration > Log Files > Execution Server Log > Location name > Execution server name

When clicking on the name of an execution server, the list of log files for the selected execution server displays. For each log file, the page displays the following columns:

| Column | Description |
|----------------|--|
| Actions | Click the buttons  and  to Delete or Download log files. |
| Name | The name of the log file. |
| Size | The physical size of the log file. |
| Date | Date when the log file was last physically saved. |

Click **Back** to return to the list of execution servers.

Administration > Log Files > Execution Server Log > Location name > Execution server name > Execution server log file name

When clicking on the name of a log file, the logging details list displays. The list includes the following items:

| Item | Description |
|-------------|--|
| Filter area | Use the filter options to filter the log list information by <i>severity</i> , <i>log level</i> , and <i>module</i> . Click Update to refresh the list according to your filter settings. |
| Table area | Displays the following logging information: Severity Severity of the event: <ul style="list-style-type: none">• Info• Warning• Error Log Level Log level of the event: <ul style="list-style-type: none">• OV = Overview• DT = Detailed• VB = Verbose• DB = Debug |

Click **Back** to return to the **Execution Server Log** page. Click **Download as CSV** to download the log file as a CSV file to your local computer.

Integrating Task Management Tools

External task management tools can integrate their tests with Silk Central through the Agile project template.

The Agile project template is a Silk Central project with the preselected attributes release and sprint. When you create a project based on the Agile project template, an empty subfolder and test container are created for the new project. The external task management tool can then insert tests into the template. For information about how to create a project based on the Agile project template, see *Adding Projects*.

The task management tool that is currently supported by Silk Central “out of the box” is VersionOne.

VersionOne is a project planning and management tool specifically designed for agile software development. It incorporates agile and iterative management practices such as release planning, iteration planning and tracking, user story or backlog item management, and task management.

Integrating VersionOne in Silk Central

To integrate tests from VersionOne into Silk Central:

1. Add a new Silk Central project and apply the Agile project template to the project.
For information on how to add a project based on the Agile template to Silk Central, see *Adding Projects*.
2. Open the file Explorer.
3. Navigate to the VersionOne configuration XML file, `VersionOneConfig.xml`.
The default path for the file is `C:\Program Files (x86)\Silk\Silk Central 20.6\instance_<instance number>_<instance name>\wwwrootAS\VersionOneIntegration`.

4. Open the VersionOne configuration XML file with an editor.
 5. Configure the VersionOne configuration XML file with the appropriate settings.
For detailed information on the settings in the VersionOne configuration XML file, refer to the comments in `VersionOneConfig.xml`.
 6. Save and close the configuration file.
-  **Note:** You do not need to restart the application server after you edit the configuration file because the file is automatically updated.
7. In Silk Central, navigate to the **Tests** area. The empty test container is displayed as incomplete, because you have to select a product.
 8. Click on the container, select the **Properties** tab, and click on the product link to browse for the product.

VersionOne is now integrated with Silk Central. New test tasks in VersionOne, for which you have defined the appropriate user, are inserted as manual tests into the default integration folder in the defined Silk Central project. The test status is now exchanged between Silk Central and VersionOne.

Refer to the `ASLogback.log` log file for information about changes to the VersionOne integration. The default path for the log file is `C:\ProgramData\SilkCentral\instance_<instance number>_<instance name>\log\`.

 **Note:** To integrate VersionOne with a Japanese Silk Central, change the start options of the Application Server service in the registry to `-Dfile.encoding=utf-8`.

Managing Products, Components, Versions, and Builds

Silk Central enables you to create and organize products, product components, versions of products, and build numbers of product versions. Product specifications can then be associated with tests, versions and builds can be associated with execution plans. You can even import products and components that were defined previously in Issue Manager.

- *Components* are discrete product elements that are tracked separately for testing purposes.
- *Versions* are product releases that are tracked separately for development and testing purposes.
- *Builds* are iterations of versions that are tracked separately for development and testing purposes.

Silk Central also enables you to set up platform designations for tests, for example operating systems. As with product, version, and build setup, platforms can be assigned names, descriptions, and active/inactive status.

Managing Builds

Builds are iterations of versions that are tracked separately for development and testing purposes.

Adding Builds

To add a new build to a version:

1. In the menu, click **Administration > Products, Versions and Builds**.
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Builds for Version** list box, select the version for which you want to add a build.
5. Click **New Build**. The **New Build** dialog box displays.
6. Type a name for the new build in the **Name** field.
7. Type a description for the build in the **Description** field.
8. Check the **Active** check box to make this build available for association with versions.

9. Click **OK** to save the build, or click **Cancel** to abort the operation.

Editing Builds

To edit an existing build within a version:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Select the build you want to edit.
5. The **Edit Build** dialog box displays, detailing the **Name**, **Description**, and **Active** status of the selected build. Make all required changes, then click **OK**.
6. Check the **Tagged** check box to tag the build.

Activating and Deactivating Builds

To activate or deactivate an existing build:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click the **Status** icon associated with the build you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
5. Confirm the message to toggle the build status to `active` or `inactive`.

Sorting the Builds List

To move a build up or down in the list or sort the builds list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Builds for Version** list box, select the version for which you want to see the builds.
5. In the **Actions** column of the item you want to move, click  or .
6. To sort the list alphabetically by name, click **Sort by Name**.

Tagging Builds

You can delete runs of tagged builds. If the selection of the runs that you are trying to delete contains tagged builds, then you get a warning and you have to confirm the action. Tagged builds will not get deleted if you perform a bulk delete (all or for a specific time span) by right clicking on the execution node.

To tag an existing build:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Builds** section, click  in the **Actions** column of the build that you want to tag. A confirmation dialog box displays.
5. Click **Yes** to continue with the tagging or click **No** to abort the tagging.

Deleting Builds

To delete an existing build from a version:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Deactivate the build you want to delete.
For additional information, see *Activating and Deactivating Builds*.
5. In the actions column of the build, click **X** . A confirmation dialog box displays.
6. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Managing Components

Components are discrete product elements that are tracked separately for testing purposes.

Adding Components

To add a new component to a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click **New Component**. The **New Component** dialog box displays.
5. Type a name for the new component in the **Name** field.
6. Type a description for the component in the **Description** field.
7. Select the **Type** of the component.
Available values are `Software` and `Documentation`.
8. Check the **Active** check box to make this component available for association with products.
9. Click **OK** to save the component, or click **Cancel** to abort the operation.

Editing Components

To edit an existing component within a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Select the component you want to edit.
5. The **Edit Component** dialog box displays, detailing the **Name**, **Description**, and **Active** status of the selected component. Make all required changes, then click **OK**.



Note: The type of a component can only be edited if you deactivate the component first. For additional information, see *Activating and Deactivating Components*.

Activating and Deactivating Components

To activate or deactivate an existing component:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click the **Status** icon associated with the component you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
5. Confirm the message to toggle the component status to `active` or `inactive`.

Sorting the Components List

To move a component up or down in the list or sort the components list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Component Type** list box, select the component type for which you want to see the components.
5. In the **Actions** column of the item you want to move, click  or .
6. To sort the list alphabetically by name, click **Sort by Name**.

Deleting Components

To delete an existing component from a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Deactivate the component you want to delete.
For additional information, see *Activating and Deactivating Components*.
5. In the actions column of the component, click . A confirmation dialog box displays.
6. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Managing Platforms

Set up platform designations for tests, for example operating systems. As with product, version, and build setup, platforms can be assigned names, descriptions, and active or inactive status.

Adding Platforms

To add a new platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. Click **New Platform**. The **New Platform** dialog box displays.
4. Type a name for the new platform in the **Name** field.
5. Type a description for the platform in the **Description** field.
6. Check the **Active** check box to make this platform available for association with tests.
7. Click **OK** to save the platform.

Editing Platforms

To edit an existing platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. In the **Actions** column of the platform that you want to edit, click .
4. The **Edit Platform** dialog box displays, detailing the **Name**, **Description**, and **Active** status of the selected platform. Make all required changes, then click **OK**.

Activating and Deactivating Platforms



Note: Platforms that are associated with a test cannot be deactivated.

To activate or deactivate an existing platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. Click the **Status** icon associated with the platform you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
4. Confirm the message to toggle the platform status to `active` or `inactive`.

Sorting the Platforms List

To move a platform up or down in the list or sort the platforms list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. In the **Actions** column of the item you want to move, click  or  .
4. To sort the platforms list alphabetically by name, click **Sort by Name**.

Deleting Platforms

To delete an existing platform:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Platforms** tab. The **Platforms** page displays, listing all existing platforms.
3. Deactivate the platform you want to delete.
For additional information, see *Activating and Deactivating Platforms*.
4. In the **Actions** column of the platform that you want to delete, click  . A confirmation dialog box displays.
5. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Platform Configuration Page

Administration > Platforms

Use this page to configure platforms.

To create a new platform, click **New Platform**. For additional information, see *Adding Platforms*.

To sort the platforms list alphabetically by name, click **Sort by Name**. For additional information, see *Sorting the Platforms List*.

For each listed platform, the page displays the following columns:

| Columns | Description |
|----------------|--|
| Actions | The user can perform the following actions on a platform: <ul style="list-style-type: none"> Moves the platform up one row in the list. Moves the platform down one row in the list. Deletes the platform permanently. Platforms need to be inactive before you can delete them, and deletion is not allowed if a platform is already associated to a test. |

| Columns | Description |
|--------------------|--|
| Name | The name of the platform as it displays in the GUI and in reports. Click the name of a platform to modify the name, description, and status of the platform. |
| Description | A textual description of the platform. |
| Status | The status of the platform, <i>Active</i> or <i>Inactive</i> . Click the status to toggle between <i>Active</i> and <i>Inactive</i> . |
| Created On | Date when the platform was created. |
| Created By | The user who created the platform. |

Managing Products

Describes how to configure products in Silk Central.

Adding Products

To add a new product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click **New Product**. The **New Product** dialog box displays.
4. Type a name for the new product in the **Name** field.
5. Type a description for the product in the **Description** field.
6. Check the **Active** check box to make this product available for association with tests.
7. Click **OK** to save the product.



Note: When creating a product, Silk Central automatically creates a new default version 1.0 and a new default build 1 for the new product.

Editing Products

To edit an existing product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. In the **Actions** column of the product that you want to edit, click . The **Edit Product** dialog box displays.
4. Make all required changes to the **Name**, **Description**, and **Active** status of the selected product, then click **OK**.

Activating and Deactivating Products

To activate or deactivate an existing product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click the **Status** icon associated with the product you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
4. Confirm the message to toggle the product status to *active* or *inactive*.

Sorting the Products List

To move a product up or down in the list or sort the products list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. In the **Actions** column of the item you want to move, click  or  .
4. To sort the list alphabetically by name, click **Sort by Name**.

Deleting Products

To delete an existing product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Deactivate the product you want to delete.
For additional information, see *Activating and Deactivating Products*.
4. In the **Actions** column of the product that you want to delete, click  . A confirmation dialog box displays.
5. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Product Configuration Page

Administration > Products, Versions, and Builds > Products

Use this page to configure products, components, versions, and builds.

Click **New Product** to create a new product. Click **Sort by Name** to sort the products alphabetically by name.

For each listed product, the page displays the following columns:

| Column | Description |
|--------------------|--|
| Actions | <p>You can perform the following actions on a product:</p> <ul style="list-style-type: none">  Moves the product up one row in the list.  Moves the product down one row in the list.  Edit the name, description, and status of the product.  Deletes the product permanently. Products need to be inactive before you can delete them, and deletion is not allowed if a product is already associated to a test. |
| Name | The name of the product as it displays in the GUI and in reports. Click the name of a product to modify the product's components, versions, and builds. |
| Description | A textual description of the product. |
| Status | The status of the product, <i>Active</i> or <i>Inactive</i> . Click the status to toggle between <i>Active</i> and <i>Inactive</i> . |
| Created On | Date when the product was created. |
| Created By | The user who created the product. |

For a selected product, the page displays the details of the included components, versions, and builds.

The page displays the following columns for the components of the product:

| Column | Description |
|----------------|--|
| Actions | You can delete the component in this column. |

| Column | Description |
|--------------------|---|
| Name | The name of the component. Click to open the Edit Component dialog box. |
| Description | The description of the component. |
| Created On | Date when the component was created. |
| Created By | The user who created the component. |
| Status | The status of the component. <i>Active</i> or <i>Inactive</i> . Click to toggle the status. |

Click **New Component** to create a new component. Click **Sort by Name** to sort the components alphabetically by name. Click **Update** to update the components list.

The page displays the following items for the versions of the product:

| Item | Description |
|--------------------|---|
| Actions | You can delete the version in this column, or move the version up or down in the list. |
| Name | The name of the version. Click to open the Edit Version dialog box. |
| Description | The description of the version. |
| Created On | Date when the version was created. |
| Created By | The user who created the version. |
| Status | The status of the version. <i>Active</i> or <i>Inactive</i> . Click to toggle the status. |

Click **New Version** to create a new version. Click **Sort by Name** to sort the versions alphabetically by name.

The page displays the following items for the builds of the product:

| Item | Description |
|--------------------|---|
| Actions | You can delete the build or tag the build. |
| Name | The name of the build. Click to open the Edit Build dialog box. |
| Description | The description of the build. |
| Created On | Date when the build was created. |
| Created By | The user who created the build. |
| Status | The status of the build. <i>Active</i> or <i>Inactive</i> . Click to toggle the status. |

Click **New Build** to create a new build. Click **Sort by Name** to sort the builds list alphabetically by name. Click **Update** to update the components list.



Note: You have to select a specific version in the **Builds** list box to enable these buttons.

Click **Back** to return to the product list.

Managing Versions

Versions are product releases that are tracked separately for development and testing purposes.

Adding Versions

To add a new version to a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Click **New Version**. The **New Version** dialog box displays.
5. Type a name for the new version in the **Name** field.
6. Type a description for the version in the **Description** field.
7. Check the **Active** check box to make this version available for association with products.
8. Click **OK** to save the version, or click **Cancel** to abort the operation.

Editing Versions

To edit an existing version within a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Select the version you want to edit. The **Edit Version** dialog box displays.
5. Make all required changes to the **Name**, **Description**, and **Active** status of the selected version, then click **OK**.

Activating and Deactivating Products

To activate or deactivate an existing product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click the **Status** icon associated with the product you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
4. Confirm the message to toggle the product status to *active* or *inactive*.

Sorting the Versions List

To move a version up or down in the list or sort the versions list alphabetically by name:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. In the **Actions** column of the item you want to move, click  or .
5. To sort the list alphabetically by name, click **Sort by Name**.

Deleting Versions

To delete an existing version from a product:

1. In the menu, click **Administration > Products, Versions and Builds** .
2. Click the **Products** tab. The **Products** page displays, listing all existing products.
3. Click a product name. The **Product Detail** page for the product opens.
4. Deactivate the version you want to delete.
For additional information, see *Activating and Deactivating Versions*.
5. In the **Actions** column of the product that you want to delete, click . A confirmation dialog box displays.

6. Click **Yes** to continue with the deletion or click **No** to abort the deletion.

Maintaining Value Lists

Value lists are used to populate custom properties within custom issue tabs that can be configured for Issue Manager in **Issues > Configuration** . For details regarding custom issue properties and tabs, see the *Issue Manager* topics in this Help.

Adding Value Lists

To add a new value list:

1. In the menu, click **Administration > Lists of Values** .
2. Click **New List**. The **New List Of Values** dialog box displays.
3. Type a name for the new value list in the **Name** field and a description for the value list in the **Description** field.
4. From the **Grouped by** list box, select a parameter by which this list is to be grouped.

This selection is definite and cannot be edited once you save your new list. Grouping values means that each value in your list must be assigned a value from the list you selected in the **Grouped by** list box. Not grouping values means that you can define custom values manually.

5. Click **New Value** to add a value to the list. The **New Value** dialog box displays.
6. Type a **Name** and **Description** for the new value.
7. Select a value from the **Grouped by** list box, if applicable.
8. Select the position of the new value within the list of existing values from the **Insert after** list box, or select `<last entry>` to add the new value to the end of the list.
9. Click **OK** to save the value. The **New List Of Values** dialog box displays the newly added value.
10. Add as many additional values as are required to complete the value list.
11. Click **OK** to save the value list.

Editing Value Lists

To edit an existing value list:

1. In the menu, click **Administration > Lists of Values** .
2. Click the name of the value list you want to edit. The **Edit List of Values** dialog box displays.
3. Change the **Name**, **Description**, and **Grouped by** setting of the selected value list, then click **OK**. If values are defined for a list, then the **Grouped by** field cannot be changed.



Note: Each individual value that is configured for the list is also available for editing. For details on editing individual values, see *Editing Individual Values*.

Editing Individual Values

To edit an individual value within a value list:

1. In the menu, click **Administration > Lists of Values** .
2. Click the name of the value list you want to edit. The **Edit List of Values** dialog box displays.
3. Click the name of the value you want to edit. The **Edit Value** dialog box displays.
4. Edit the **Name** and **Description** of the value as required.
5. Click **OK** to save your changes.

Activating and Deactivating Value Lists



Note: You cannot deactivate the value list *Issue Types*, as this list is required by Issue Manager.

To activate or deactivate an existing value list:

1. In the menu, click **Administration > Lists of Values** .
2. Click the **Status** icon associated with the value list you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
3. Confirm the message to toggle the value list status to *Active* or *Inactive*.



Note: Each individual value that is configured for the list can also be activated or deactivated. For details on activating or deactivating individual values, see *Activating and Deactivating Individual Values*.

Activating and Deactivating Individual Values

To activate or deactivate an individual value within a value list:

1. In the menu, click **Administration > Lists of Values** .
2. Click the name of the value list that includes the value you want to activate or deactivate. The **Edit List of Values** dialog box displays.
3. Click the **Status** icon associated with the value you want to activate or deactivate. A confirmation dialog box displays.
4. Click **OK** to save the updated value list.

Sorting Values within Value Lists

To move a value up or down within a value list or sort the value list alphabetically by value name:

1. In the menu, click **Administration > Lists of Values** .
2. Click the name of the value list that includes the value you want to sort. The **Edit List of Values** dialog box displays.
3. In the **Actions** column of the item you want to move, click  or .
4. To sort the value list alphabetically by value name, click **Sort by Value**. The **Sort By Value** dialog box displays.
5. Select the sort order, *Ascending* or *Descending* and click **OK**.
6. Click **OK** on the **Edit List Of Values** dialog box to save the updated value list.

Deleting Value Lists

To delete an existing value list:

1. In the menu, click **Administration > Lists of Values** .
2. Deactivate the value list you want to delete.
For additional information, see [Activating and Deactivating Value Lists](#).
3. In the Actions column of the value list that you want to delete, click . A confirmation dialog box displays.
4. Click **Yes** to continue with the deletion or click **No** to abort the deletion.



Note: Each individual value that has been configured for the list is also available for deletion. For details on deleting individual values, see [Deleting Individual Values](#).

Deleting Individual Values

To delete an individual value within a value list:

1. In the menu, click **Administration > Lists of Values**.
2. Click the name of the value list that includes the value you want to delete. The **Edit List of Values** dialog box displays.
3. Deactivate the value you want to delete. For additional information, see [Activating and Deactivating Individual Values](#).
4. In the **Actions** column of the value that you want to delete, click **X**.
5. Click **OK** on the **Edit List Of Values** dialog box to save the updated value list.

List of Values Configuration Page

Administration > Lists of Values

Use this page to configure value lists. For each listed list of values, the page displays the following columns:

| Column | Description |
|-------------------|--|
| Name | The name of the value list as it displays in the GUI and in reports. Click the name of a value list to modify the name, description, and values of the value list. |
| Grouped By | Displays if a value list is grouped by another list. |
| Status | The status of the value list, <i>Active</i> or <i>Inactive</i> . Click the status to toggle the between <i>Active</i> and <i>Inactive</i> . The default value list, <i>Issue Types</i> , cannot be deactivated, because it is a key component used in Issue Manager. |
| Created On | Date when the value list was created. |
| Created By | The user who created the value list. |
| Changed On | Date when the value list was last modified. |
| Changed By | The user who last modified the value list. |
| Actions | You can perform the following actions on a value list: <ul style="list-style-type: none"> X Deletes the value list permanently. Value lists need to be inactive before you can delete them, and deletion is not allowed if a value list is already associated with issues. |

Click **New List** to create a new value list.

Configuring Global Schedules

You can define global schedules that can be re-used in Silk Central for the scheduling of execution plans. Global schedules speed up the process of scheduling execution plans since the need to define individual schedules is reduced to only those execution plans that require special scheduling. For detailed information on applying schedules, see [Schedules](#).

Exclusions

You can define weekdays and time-of-day intervals during which tests are not to be executed. For example, you may not want tests to take place on weekends.

Definite Runs

You can define points in time at which tests must execute, regardless of configured schedules.

Adding Global Schedules

To add a new global schedule:

1. In the menu, click **Administration > Schedules** . The **Schedules** page displays, listing all existing global schedules.
2. Click **New Schedule**. The **Configure Schedule** page displays.
3. Type a name for the new schedule in the **Name** field. It is good practice to include the time zone in the name, for example "Daily at 6 PM EST", so that users know when it actually runs.
4. Type a description for the schedule in the **Description** field.
5. Click  next to **From** to define a start date and time for the schedule. All date and time values are saved in GMT to the database. The presentation of values is set based on the **Time zone** setting specified in the user settings. For example, if an administrator who is located in New York creates a global schedule that runs every day at 6 PM, it runs at midnight for a user located in Paris, who applies this schedule to his execution plan.
6. Define the **Interval** at which an execution should repeat. The available time range is from 1 minute to 36 days.
7. Check the **Adjust schedule to daylight savings** check box to have scheduled executions automatically adjust to daylight-saving time changes.
8. Click  next to run to define the date and time at which execution should end. Click **Forever** to execute for an undefined time period, or **until** to execute for a defined period.
9. To define weekdays and time-of-day intervals at which the test should not execute, click **Add Exclusion**.
For additional information, see *Scheduling Exclusions*.
10. To define a point in time when the test must execute, click **Add Definite Run**.
For additional information, see *Scheduling Definite Runs*.
11. Click **Save** when you have finished defining the schedule.

Editing Global Schedules

To modify a global schedule:

1. In the menu, click **Administration > Schedules** . The **Schedules** page displays, listing all existing global schedules.
2. Click the name of the schedule you want to edit. The **Configure Schedule** page displays.
3. Change the name of the schedule in the **Name** field. It is good practice to include the time zone in the name, for example "Daily at 6 PM EST", so that users know when it actually runs.
4. Change the description of the schedule in the **Description** field.
5. Click  next to **From** to define a new start date and time for the schedule. All date and time values are saved in GMT to the database. The presentation of values is set based on the **Time zone** setting specified in the user settings. For example, if an administrator who is located in New York creates a global schedule that runs every day at 6 PM, it runs at midnight for a user located in Paris, who applies this schedule to his execution plan.
6. Change the **Interval** at which an execution should repeat. The available time range is from 1 minute to 36 days.
7. Check the **Adjust schedule to daylight savings** check box to have scheduled executions automatically adjust to daylight-saving time changes.

8. Click  next to run to define the date and time at which execution should end. Click **Forever** to execute for an undefined time period, or **until** to execute for a defined period.
9. To define weekdays and time-of-day intervals at which the test should not execute, click **Add Exclusion**.
For additional information, see *Scheduling Exclusions*.
10. To define a point in time when the test must execute, click **Add Definite Run**.
For additional information, see *Scheduling Definite Runs*.
11. Click **Save** when you have finished modifying the schedule.

Activating and Deactivating Global Schedules

To activate or deactivate an existing product:

1. In the menu, click **Administration > Schedules**. The **Schedules** page displays, listing all existing global schedules.
2. Click the **Status** icon associated with the schedule that you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
3. Confirm the message to toggle the schedule status to *Active* or *Inactive*.

Deleting Global Schedules

To delete an existing global schedule:

1. In the menu, click **Administration > Schedules**. The **Schedules** page displays, listing all existing global schedules.
2. In the **Actions** column of the schedule that you want to delete, click . A confirmation dialog box displays.
3. Click **Yes** to remove the current schedule, including all exclusions and definite runs, or click **No** to abort the deletion.



Note: Once a global schedule is in use by a test, you cannot delete it. You must first re-schedule all tests to another schedule.

Scheduling Exclusions

To schedule exclusions:

1. In the menu, click **Administration > Schedules**. The **Schedules** page displays, listing all existing global schedules.
2. Click the name of the schedule you want to edit.
3. On the **Configure Schedule** page, click **Add Exclusion**.
4. On the **Configure Schedule Exclusion** page, select the weekdays on which tests should be suppressed.
5. Define the specific time intervals on those days during which execution should be suppressed.
6. Click **OK** once you have completed definition of exclusion settings, or click **Cancel** to abort. Exclusion settings are now listed on the **Configure Schedule** page.
7. Click **Save** to add the exclusion to the current schedule, or continue adding additional exclusions.



Tip: You can edit and delete exclusions by clicking  or  in the **Actions** column of the exclusion that you want to modify.

Scheduling Definite Runs

To schedule definite runs:

1. In the menu, click **Administration > Schedules** . The **Schedules** page displays, listing all existing global schedules.
2. Click the name of the schedule you want to edit.
3. On the **Configure Schedule** page, click **Add Definite Run**.
4. On the **Configure Definite Run** page, click  to define a point in time for the definite run.
5. Click **OK** once you have completed definition of the definite run, or click **Cancel** to abort. Definite run settings are now listed on the **Configure Schedule** page.
6. Click **Save** to add the definite run to the current schedule, or continue adding additional definite runs.



Tip: You can edit and delete definite runs by clicking  or  in the **Actions** column of the definite run that you want to modify.

Schedules Configuration Page

Administration > Schedules

Use this page to configure global schedules for Silk Central. For each schedule, the page displays the following columns:

| Column | Description |
|--------------------|---|
| Actions | This column contains action icons which allow the user to perform the following actions on a schedule:  Deletes the schedule permanently. Deletion is not allowed if a schedule is already associated with execution plans. |
| Name | The name of the schedule as it displays in the GUI and in reports. Click the name of a schedule to modify the name, description, and status of the schedule. |
| Description | A textual description of the schedule. |
| Status | The status of the schedule, <i>Active</i> or <i>Inactive</i> . Click the status to toggle between <i>Active</i> and <i>Inactive</i> . |
| Created On | Date when the schedule was created. |
| Created By | The user who created the schedule. |

Click **New Schedule** to create a new schedule.

Time Zones

Silk Central is designed to execute tests over a network of execution servers. Because the Internet enables such networks to be spread worldwide across multiple time zones, it is important to understand time-zone handling in Silk Central.

All date and time values are saved in GMT to the database. The presentation of values is set based on the **Time zone** setting specified in the user settings. This needs to be considered especially when you create globally usable schedules. For example, if an administrator who is located in New York creates a global schedule that runs every day at 6 PM, it runs at midnight for a user located in Paris, who applies this schedule to his execution plan. It is good practice to include the time zone in the name, for example "Daily at 6 PM EST", so that users know when it actually runs.

For information on time zone settings, see *Adding User Accounts*.

The following requirements apply:

- The application server and front-end server should be in the same time zone. Separating these servers locally within a WAN does not make sense because the application server communicates closely with one or several front-end servers. Also, front-end servers as well as the application server have direct database access.
- Execution servers may be in different time zones, separated both from the application server and from other execution servers.

Managing Projects

Projects in Silk Central usually reflect a software project in your company, respectively the work of a development team, with a common release date, a common source control system, common requirements, etc.

The **Projects** area offers a high-level test manager's view of all projects in your Silk Central installation, including a list of the most recent projects you have set to active, and enables you to move between projects. If you have SuperUser, Administrator, or Project Manager privileges, you can specify project-wide settings for Silk Central projects. Once global project settings are defined, they are available to all users who have access to those projects.

Working with Projects

This topic describes the conceptual background of projects in Silk Central.

Silk Central manages requirements, tests, their execution and the results in the context of a project. Therefore the first thing you have to do is to create a project. Projects serve as containers for related sets of tasks and results. Resources such as project managers and analysts are allocated to projects by assigning them to user groups, which have access rights to certain projects.

Adding Projects

To create a project:

1. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
2. Click **New Project**. The **Project Settings** page displays.
3. Type a **Project name** and **Description**.
4. Select the **Project Owner**.
5. *Optional:* Enter **Tags** for your project. If you have many projects, project tags can help you filter them, using the **Tags** column on the **Project List** page.
6. *Optional:* To create a project based on the Agile project template, choose *Agile Project Template* from the **Project Template** list box.
7. To initialize the project as an Issue Manager project, click **Initialize new project** or **Reuse existing project**. For more information, see *Initializing Projects as Issue Manager Projects*.
8. In the **Shared Tests** section, define whether the project is sharing tests or not. Select one of the following options:
 - **No test sharing:** This project neither shares tests with other projects nor consumes tests from other projects.
 - **Sharing tests:** This project is sharing tests that other projects can consume.
 - **Consuming shared tests from:** Select a project from which the edited project consumes shared tests.

9. The **Groups** section includes lists of available and assigned groups. Assign the groups as required. Use **Ctrl+Click** or **Shift+Click** to select multiple groups. You can also import groups from an LDAP server by clicking **Import LDAP Group**. For detailed information, see *Importing Groups from LDAP Servers*. If no user groups exist, you may assign them later after you have created them. You can also configure the group/project assignment on the **Group Settings** page. Privileges vary based on user roles. For information about user privileges, see *User Roles and Permissions*.
10. The **Locations** section includes lists of available and assigned locations. Assign the locations as required. Use **Ctrl+Click** or **Shift+Click** to select multiple locations. If no locations exist, you may assign them later after you have created them. You can also configure the location/project assignment on the **Location Settings** page. For detailed information, see *Managing Locations*.
11. Click **Save** to save your settings. You are returned to the **Project List** page where the new project is listed.

Project Baselines

This topic describes the conceptual background of project baselines in Silk Central.

A project baseline is a snapshot of a project at a given time and can be created for any project or even another project baseline. The last run of each execution plan in the project is also included in the project baseline. The schedule options included in a project baseline are initially set to none, to prevent losing the execution status of the last runs. An example usage of a project baseline is to save a snapshot of a project immediately after a release. The *Baseline Comparison* report in Silk Central shows the amount of tests that are currently different in the project compared to the project baseline. For more information on the report, refer to the *Silk Central Help*.

For each new project baseline that includes an element of the **Tests** area, the **History** page of the element includes an entry with links to the project baseline and the corresponding element in the project baseline. If the element itself was created as part of a project baseline, the first entry in the **History** page includes links to the original project and the corresponding element in the original project.

When tests in the original project include calls to shared steps objects or keywords from libraries, you can define during the creation of the project baseline whether all calls are resolved or kept. The project baseline has the same visibility on the libraries as the original project. For more information on shared steps objects and keywords, refer to *Libraries* in the *Silk Central Help*.



Note: Project baselines can be created by all users that have the **Manage projects** permission.

Creating a Baseline for a Project



Caution: Creating a baseline for a project can lock the database for several minutes, depending on the size of the project that the baseline is created for. We recommend to create a baseline for a project during off-hours, when user activity on Silk Central is minimal.



Note: If the original project includes schedules, the scheduling options in the new baseline are set to none to prevent losing the last execution status. When you create a baseline, the **History** page of each **Test** item included in the baseline is updated with an entry for the baseline.

To create a baseline for a project:

1. In the menu, click **Projects > Project List**. The **Projects** page displays, listing all existing projects and project baselines.
2. In the **Actions** column of the project you want to create a baseline for, click . The **Baseline Project** dialog box displays.
3. Type a name and a description for the new baseline.
4. *Optional:* If the project contains tests that use shared steps or keywords from libraries, define the handling of the shared steps in the **Library assets handling** section:
 - Click **Copy library assets** to detach all references to shared steps and keywords in the original libraries, and instead reference assets in a newly created copy of these libraries in the baseline.

Choose this option to create a project baseline for historical or auditing purposes. The library copies are read-only by default.

- Click **Keep references to library assets** to keep all references to shared steps and keywords in the original libraries. Choose this option to create a working copy for ongoing testing purposes, for example a new version of the project.
 - To prevent users from editing the library, set the library as read-only. For additional information, see [Managing Keyword Libraries and Library Properties](#).
5. *Optional:* If the project contains manual or keyword-driven tests, define in the **Test asset versioning** section whether a new version should be created for the tests in the original project and/or in the baseline.
- Check the **Create new versions for manual and keyword-driven assets in original project** check box to create new versions for all manual and keyword-driven tests with shared objects in the original project.
 - Check the **Create new versions for manual and keyword-driven assets in new baseline** check box to create new versions for all manual and keyword-driven tests with shared objects in the baseline. If you selected **Copy library assets**, new versions are also created for all assets in the copied libraries.



Note: Versions are not created for data-driven instances.

6. Click **OK**. The **Baseline Project** dialog box displays and informs you that the operation may take several minutes to complete.
7. Click **Yes** to continue. The **Baseline Project - Adapt Project Settings** dialog box opens. The dialog box displays the requirement integration settings along with all source control profiles that are configured for the original project.
8. Click the corresponding **Edit** button to link the project baseline to the appropriate baseline, branch, label, or other point of reference in the source control profile, or to change the requirement integration settings.



Note: For more information on editing source control profiles, refer to the *Silk Central Help*.

9. Click **Finish** to finish creating the baseline.
10. Click **Yes** in the **Baseline Project** dialog box to activate the new baseline.

Editing Projects and Project Baselines

To edit an existing project or project baseline:

1. In the menu, click **Projects > Project List**. The **Projects** page displays, listing all existing projects and project baselines.
2. Click  in the Actions column of the project name of the project or project baseline you want to edit.



Note: The project or project baseline must be inactive.

3. Edit the **Project name** and **Description** as required.
4. Change the **Project Owner** as required.
5. Check the **Active** check box to activate the project or project baseline.
6. *Optional:* Enter **Tags** for your project. If you have many projects, project tags can help you filter them, using the **Tags** column on the **Project List** page.
7. In the **Shared Tests** section, define whether the project is sharing tests or not. Select one of the following options:
 - **No test sharing:** This project neither shares tests with other projects nor consumes tests from other projects.

- **Sharing tests:** This project is sharing tests that other projects can consume.
 - **Consuming shared tests from:** Select a project from which the edited project consumes shared tests.
8. The **Groups** section includes lists of available and assigned groups. Assign the groups as required. Use **Ctrl+Click** or **Shift+Click** to select multiple groups. You can also import groups from an LDAP server by clicking **Import LDAP Group**. For detailed information, see *Importing Groups from LDAP Servers*. If no user groups exist, you may assign them later after you have created them. You can also configure the group/project assignment on the **Group Settings** page. Privileges vary based on user roles. For information about user privileges, see *User Roles and Permissions*.
 9. The **Locations** section includes lists of available and assigned locations. Assign the locations as required. Use **Ctrl+Click** or **Shift+Click** to select multiple locations. If no locations exist, you may assign them later after you have created them. You can also configure the location/project assignment on the **Location Settings** page. For detailed information, see *Managing Locations*.
 10. Click **Save** to save your settings. You are returned to the **Project List** page.

Activating or Deactivating Projects and Project Baselines



Note: You can also activate or deactivate an existing project or project baseline from the **Project List** page. For additional information, see *Editing Projects*.

To activate or deactivate an existing project or project baseline:

1. In the menu, click **Projects > Project List**. The **Projects** page displays, listing all existing projects and project baselines.
2. Click **Active/Inactive** in the **Status** column of the project or project baseline you want to activate or deactivate. A confirmation dialog box displays, asking you if you are sure about the activation or deactivation.
3. Confirm to toggle the project status to **Active** or **Inactive**.

Copying Projects or Project Baselines

Copy an existing project or project baseline to a new project.

If the original project or baseline includes schedules, the scheduling options are set to none during the copy.



Caution: Copying a project or project baseline can lock the database for several minutes, depending on the size of the project or project baseline that is being copied. It is recommended to copy projects or project baselines during off-hours, when user activity on Silk Central is minimal.



Tip: Especially if large projects are written to the database, this can significantly slow down performance of the database. We strongly recommend to do a full index rebuild for the Silk Central database schema within your next maintenance window.

To copy a project or project baseline:

1. In the menu, click **Projects > Project List**. The **Projects** page displays, listing all existing projects and project baselines.
2. Click  in the **Actions** column of the project or project baseline you want to copy. The **Copy Project** dialog box displays.
3. In the **New project name** field, type a name for the new project.
4. Check the check boxes of any additional information types you want to have copied along with the new project. When you check the **Test history** check box, the versions are also copied.
For detailed information on the check boxes, see *Copy Project Dialog Box*.
5. Click **OK**.
6. If no Issue Manager project is initialized for the project you are copying, go on with step 10. If an Issue Manager project is initialized, a dialog box appears:

- If you want to reuse the existing Issue Manager project, click **Yes** and go on with step 10.
- If you do not want to reuse the existing Issue Manager project but if you want to copy it, click **No** and go on with step 7.

Independent of whether you choose to reuse the existing Issue Manager project or not, any copied issue tracking profiles will continue to point to the original Issue Manager project.

7. Check the **Copy Issues** check box to copy all of the source project's existing issues to the new project.
8. Check **Copy Archived Issues** to copy all of the source project's archived issues to the new project.
9. Click **OK**.
10. Click **Yes** on the **Copy Project** dialog box, confirming that you want to begin the copy process and that you know the process may take several minutes. When the copy process is complete, a dialog box informs you what has been copied and asks you if you want to activate the project, thereby making the project available in Silk Central's project view.

Deleting Projects and Project Baselines

 **Caution:** When you delete a project or project baseline you permanently remove all related results from the repository. You also destroy all content associated with the project or project baseline. If you want to keep results, we recommend that you set a project or project baseline to inactive rather than delete it. For information on deactivating projects and project baselines, see *Activating or Deactivating Projects or Project Baselines*.

To delete a project or project baseline:

1. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
2. Click  in the **Actions** column of the project or project baseline you want to remove.

 **Note:** The project or project baseline must be inactive.

A confirmation dialog box displays, asking you to confirm the deletion.

3. Click **Yes** to remove the project or project baseline; or click **No** to abort the operation. If you choose **Yes**, you will be returned to projects list, where the deleted project or project baseline is no longer listed.

Exporting Projects

You can export a project including all the project data as a .zip file to your local computer. This is especially useful if you want to archive the data outside of Silk Central.

To export a project:

1. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
2. Click  (**Export project**) in the **Actions** column.
3. Click **OK** on the confirmation dialog box.
4. Specify a location on your computer to save the project data there. Silk Central saves the project data in a .zip file.

 **Attention:** Before you start to export or import projects, carefully read the recommendations below.

- When you export/import large projects with a huge amount of project data, the export/import process can take a long time (up to some hours).
- The file size can vary tremendously from project to project and depends on many factors, for example: On the amount of assets, the run count, the attachments, the code coverage statistics, and most importantly on the result files.

- We recommend to minimize the size of a project before you export it. Therefore, you should remove unneeded data and especially unused result files. If your project exceeds the size of 1.2 GB, you will receive the error `Request entity is too large` when attempting to import it. If you cannot reduce the size of the project anymore, you can increase the upload limit in the file `nginx.conf.template` in `C:\ProgramData\SilkCentral\InstanceAdministration\nginx\conf` by modifying the value of the `client_max_body_size` property.
- Especially if large projects are written to the database, this can significantly slow down performance of the database. We strongly recommend to do a full index rebuild for the Silk Central database schema within your next maintenance window.
- Some browsers have file upload limitations, like for example Internet Explorer. If the project that you want to import exceeds that size, the import will fail. Use another browser to avoid this issue.
- Be aware that the export/import process puts considerable load on Silk Central and on the database server. Export/import actions should be done during off-hours (for example over night or on the weekend) or at least when the Silk Central usage is low and no users are logged in.
- Exporting and importing data is tightly bound to and relying on the database and its content. If the database already contains any inconsistency in data, importing a project might fail. Hence, we strongly recommend - as good practice for every backup process - to try a reimport immediately after an export to ensure validity, data consistence and recoverability.
- If you want to reimport project data, use the exact same .zip file that you have exported. Other file formats will not be accepted. You can import the data only into the same Silk Central and database version that you have used to export the data.
- If you have any questions concerning the export/import process, do not hesitate to contact customer support.

Importing Projects

Silk Central projects can be exported to archive the project data outside of Silk Central. To view the data, all results, reports, and so on, you have to reimport the project.

When you import project data, you must use the same Silk Central version and the same database version that were used to export the data. However, you can import project data to the same or to a different database server with the same version that was used to export the data.

To import a project:

1. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
2. Click **Import Project** on the top. The **Import Project** dialog box appears.
3. Browse for an appropriate .zip file on your computer and click **OK**. Silk Central uploads the data.



Note: The .zip file has to be a file that was exported from Silk Central beforehand.



Note: The .zip file is uploaded to the front-end server before it is imported into the database. Therefore, the front-end server needs to provide appropriate free disk space to store the .zip file.

4. Click **Yes** to start the import. Note the following:
 - Silk Central processes only one import at a time.
 - Silk Central writes audit logs when an import starts, finishes or fails.
 - The import is processed in the background. You are notified by email when the import is complete. Make sure that you have an email address and an email server configured before you start the import. Otherwise you will not be notified. For more information, see *Configuring Email Servers* and *User Settings Page*.
 - The project appears in the project list when the import is complete.



Attention: Before you start to export or import projects, carefully read the recommendations below.

- When you export/import large projects with a huge amount of project data, the export/import process can take a long time (up to some hours).
- The file size can vary tremendously from project to project and depends on many factors, for example: On the amount of assets, the run count, the attachments, the code coverage statistics, and most importantly on the result files.
- We recommend to minimize the size of a project before you export it. Therefore, you should remove unneeded data and especially unused result files. If your project exceeds the size of 1.2 GB, you will receive the error `Request entity is too large` when attempting to import it. If you cannot reduce the size of the project anymore, you can increase the upload limit in the file `nginx.conf.template` in `C:\ProgramData\SilkCentral\InstanceAdministration\nginx\conf` by modifying the value of the `client_max_body_size` property.
- Especially if large projects are written to the database, this can significantly slow down performance of the database. We strongly recommend to do a full index rebuild for the Silk Central database schema within your next maintenance window.
- Some browsers have file upload limitations, like for example Internet Explorer. If the project that you want to import exceeds that size, the import will fail. Use another browser to avoid this issue.
- Be aware that the export/import process puts considerable load on Silk Central and on the database server. Export/import actions should be done during off-hours (for example over night or on the weekend) or at least when the Silk Central usage is low and no users are logged in.
- Exporting and importing data is tightly bound to and relying on the database and its content. If the database already contains any inconsistency in data, importing a project might fail. Hence, we strongly recommend - as good practice for every backup process - to try a reimport immediately after an export to ensure validity, data consistence and recoverability.
- If you want to reimport project data, use the exact same .zip file that you have exported. Other file formats will not be accepted. You can import the data only into the same Silk Central and database version that you have used to export the data.
- If you have any questions concerning the export/import process, do not hesitate to contact customer support.

Copy Project Dialog Box

Projects > Projects List > Copy Project

Use this dialog box to copy an existing project to a new project.



Note: Check the check boxes of any additional information types you want to have copied along with the new project. If you don't select additional information types, an empty project with only group assignments, location assignments, and project settings is copied.

| Item | Description |
|---|---|
| New project name | Specifies the name of the new project to which the existing project is copied to. |
| New project description | Specifies the description of the new project to which the existing project is copied to. |
| Settings | Check this check box to copy project settings, filters, attributes, custom requirement properties, change notification, and integration configuration setting information to the new project. Third-party integration configuration setting information is not copied. |
| Requirements tree (includes Requirements settings) | Check this check box to copy the requirements tree with all information for each requirement, for example properties, attachments, and others. Third-party requirement information is not copied. When a project has been configured for an external requirements management system, you are asked if you want to transfer the RMS settings to the copy of the project. |
| Requirements history | Check this check box to copy the historical information for each requirement. This option is only available when Requirements tree is checked. |

| Item | Description |
|---|---|
| Tests tree (includes Tests settings) | Check this check box to copy the tests tree with all information for each test. When both Requirements tree and Tests tree are checked, in addition to the information listed above, the information related to the relationship between requirements and tests (assigned requirements and assigned tests) is also copied |
| Tests history | Check this check box to copy the historical information for each test. This option is only available when Tests tree is checked. |
| Execution Plans tree (includes Tests tree) | Check this check box to copy the execution tree with all information for each execution plan. |
| Last runs of tests | Check this check box to copy the last run of each test. This option is only available when Execution Plans tree is checked. When Requirements tree , Execution Plans tree , and Last runs of tests are checked, all trees are copied in their entirety. The relationship between the trees is also established. |
| Reports | Check this check box to copy all project-related reports to the new project. |

For Issue Manager, the dialog box displays the following additional check boxes:

| Copy Project — Issue Manager Dialog Item | Description |
|--|--|
| Copy Issues | Check this check box to copy all of the source project's existing issues to the new project. |
| Copy Archived Issues | Check this check box to copy all of the source project's archived issues to the new project. |

If you do not check either check box, issue data will not be copied to the new project.

Source project configurations are automatically copied to the new project. These include:

- Products
- GUI configuration
- Notification rules and project-wide notifications
- Routing rules
- Workflow
- User views on inboxes
- Workgroups
- Inboxes
- User account configuration

Project Settings Page

Project > New Project

Use the **Project Settings** page to configure projects. The page displays the following items:

| Item | Description |
|----------------------|---|
| Project Name | Specifies the name of the project as it should appear in the GUI and in reports. |
| Description | A description of the project. You can enter any text for the description. |
| Project Owner | Specifies the owner of the project. The selected user account does not have any special privileges; this setting is purely informative. |
| Active | Check this check box to activate the project. Inactive projects are not visible in your application. |

| Item | Description |
|----------------------------------|---|
| Tags | If you have many projects, project tags can help you filter them, using the Tags column on the Project List page. |
| Project Template | This list box defines whether a project is based on the Agile project template or not. Choose Agile Project Template to base the project on the template for Agile project-management tools, or <None> to create a simple project. You can only specify this while creating a new project, not while editing an existing project. |
| Issue Manager Integration | Select one of the following options: <ul style="list-style-type: none"> • No project: No Issue Manager integration is created. • Initialize new project: The new project is initialized as an Issue Manager project. When you click Save on the bottom of the page, a dialog box appears, where you can configure all necessary settings for the project (groups, inboxes, products, and so on). • Reuse existing project: An existing project is used. Select a project from the list. If the project was not configured as an Issue Manager project before, you have to configure all necessary settings. A dialog box appears, when you click Save. If the project was already configured as an Issue Manager project before, you have to configure the credentials and the status mapping only. |
| Shared Tests | Select one of the following options: <ul style="list-style-type: none"> • No test sharing: This project neither shares tests with other projects nor consumes tests from other projects. • Sharing tests: This project is sharing tests that other projects can consume. • Consuming shared tests from: Select a project from which the edited project consumes shared tests. |
| Groups | The Groups section includes lists of available and assigned groups. Assign the groups as required. Use Ctrl+Click or Shift+Click to select multiple groups. You can also import groups from an LDAP server by clicking Import LDAP Group . For detailed information, see <i>Importing Groups from LDAP Servers</i> . If no user groups exist, you may assign them later after you have created them. You can also configure the group/project assignment on the Group Settings page. Privileges vary based on user roles. For information about user privileges, see <i>User Roles and Permissions</i> . |
| Location | The Locations section includes lists of available and assigned locations. Assign the locations as required. Use Ctrl+Click or Shift+Click to select multiple locations. If no locations exist, you may assign them later after you have created them. You can also configure the location/project assignment on the Location Settings page. For detailed information, see <i>Managing Locations</i> . |

Initializing Projects as Issue Manager Projects

When you create a new project in Silk Central, you can integrate the project with an Issue Manager project:

1. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
2. Click **New Project**. The **Project Settings** page displays.
3. Configure the settings. For more information, see *Adding Projects*.
4. In the **Issue Manager Integration** row, you can select:
 - **Initialize new project:** Initializes a new Issue Manager project and integrates it with the newly created Silk Central project.

- **Reuse existing project:** An existing Issue Manager project is integrated with the newly created Silk Central project. Select a project from the list. You will have to configure the credentials and the status mapping.
5. Click **Save** on the bottom of the page. The new project is created and appears in the **Projects List**. The **Configure Issue Manager Project** dialog box appears.
 6. Configure your group settings and click **Next**.
 7. You can manually add inboxes or let Silk Central generate inboxes for you:
 - Click **Add Inbox** to manually add inboxes.
 - Click **Generate Inboxes** to let Silk Central generate inboxes for all users that are assigned to the project. Check **Use full name for generation** to use the full name of a user as inbox name instead of the login name of the user. Check **Auto generate inboxes in future** to let Silk Central automatically generate inboxes for users and usergroups that are assigned in the future.
 8. Define your product settings and click **Close**.
 9. Provide default credentials and click **OK**. The user name you enter needs to be part of the user group you assigned to the project.
 10. Configure how the external and internal properties shall be mapped and click **OK**.



Note: You can cancel the initializing process at any time. However, to complete the initialization later, you have to configure these settings on the **Issue Tracking Profile** page, the **Issue Manager Project List** page, and the **Issue Manager Configuration** page.

Build Information

Build information files contain version and build information that is used for execution runs. The build information files are typically stored and searched for on the execution server that is executing the execution run. If a build information file is not found on the execution server, the file is searched for on the application server. This behavior is beneficial when you have multiple execution servers and use a single build information file across all execution servers. You need to maintain only a single build information file on the application server. The build information files for automated tests that have no execution server assigned, and for all manual tests, are immediately searched for on the application server.

Silk Central is able to match up test results with build information and display test results for specific build numbers.

Build Information Updates

Build information files must be created and configured manually. Whenever a new build becomes available for testing on an execution server, update the build information to reflect the new build number.

You can update the build information in two different ways:

- Manually, by editing the files each time a new build is installed.
- Automatically, if you are using an automated build update process to update the build information file, for example through a VB Script.

Creating Build Information Files

To create a build information file:

1. On the application server, navigate to: C:\ProgramData\SilkCentral\instance_<instance number>_\<instance name>\BuildInfos.
2. Create a build info file for your project based on the template file `BuildInfoExample.xml`. The template file contains the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<ProjectBuildInfo>
  <BuildEntryList>
```

```

    <BuildEntry name="Demo Product">
      <Version>3.1</Version>
      <BuildNr>350</BuildNr>
    </BuildEntry>
    <BuildEntry name="Product2">
      <Version>4.2</Version>
      <BuildNr>613</BuildNr>
    </BuildEntry>
  </BuildEntryList>
</ProjectBuildInfo>

```



Note: An element called `BuildEntryList` contains a list of `BuildEntry` elements. `BuildEntry` tags refer to specific products that are defined by the name attribute of `BuildEntry` elements.

3. Modify the file content to fit your environment.

Version The number of the version that is currently available for testing. The number is not necessarily the same for each execution server.

BuildNr The number of the build that is currently available for testing. The number is not necessarily the same for each execution server.

4. *Optional:* If you require specific execution servers to use different version and build information, distribute the modified build information file to the execution server(s) at `C:\ProgramData\SilkCentral\BuildInfos`. For Linux execution servers, the file needs to be stored in the sibling-folder `../ExecServerData` of the directory from which the execution server is started.

5. Once you have created the build information files, you must specify the file name in the settings of the corresponding project. Click **Projects:<Project Name> > Projects List** to view the list of projects assigned to you. Select the project to which you want to link the build information.

in the sub-folder `ExecServerData` of the current directory



Note: This must be done before the scheduling of any tests for the project. Otherwise previously scheduled tests will not be updated.

6. Click the **Project Settings** tab.

7. Click **Edit** to edit the project settings of the selected project. The **Edit Project Settings** dialog box opens.

8. Specify the name of the previously created XML file in the **Build information file name** field.

9. Click **OK** to update the information. With all future test executions, Silk Central will read build information from the corresponding file and match test results with that information.

Comparing a Project with a Baseline

For baselined projects, you can use the *Baseline Comparison* report to see all the changes to the tests in the project since the moment the project was baselined.

To view the report:

1. In the menu, click **Projects > Project List**.

2. Use **Ctrl+Click** to select the project and the baseline of the project that you want to compare.



Tip: For easier selection of the baselines, don't click on the name of the baseline. Click into another column instead.

3. Right click on the selection.

4. Select **Reports > Baseline Tests Comparison**.



Note: The *Baseline Comparison* report must be executed once to display in the context-menu. If the report is grayed out, execute the report once in the **Reports** area.

Selecting Projects

To select a project:

1. In the menu, click **Projects > Project List**.
2. Click the name of the project to select it.



Note: You can only select active projects.

Project List

Project:<Project Name> > Project List

The **Project List** page alphabetically displays all the projects and project baselines associated with your Silk Central installation. The page supports filtering, but does not support sorting. The displayed projects are sorted by name. Inactive projects and inactive project baselines are disabled.

Project baselines are sorted by their creation date, marked with a special icon, and shown as child nodes of the original project.

For more information, see *Managing Projects*.

The following columns are displayed for each project and project baseline:

| Column | Description |
|---------------|---|
| Actions | <ul style="list-style-type: none"> Delete Edit project Copy project Baseline project Export project <p> Note: This column will be hidden if the current user does not have modify/delete project permissions.</p> |
| Project | Name of the project. Click to set the project as the active project. |
| Project ID | The identifier of the project. |
| Status | <i>Active</i> or <i>Inactive</i> . Click to switch the status. |
| Description | Project description. |
| Project Owner | Specifies the owner of the project. The selected user account does not have any special privileges; this setting is purely informative. |
| Release Date | Specifies the planned release date for the project. |
| Created On | Date and time the project was created. |
| Created By | User who created the project. |
| Changed On | Date and time the project was last changed. |
| Changed By | User who last changed the project. |

| Column | Description |
|--------|---|
| Tags | <p>If you have many projects, project tags can help you filter them, using the Tags column on the Project List page. Use Ctrl+Click or Shift+Click to select multiple projects, then right-click your selection and click Edit Tags to manage the tags of multiple projects at once.</p> <p> Note: Filtering by tags uses an OR condition, meaning that all projects are displayed that contain either of the selected tags.</p> |

Switching to a Recently-Accessed Project

You can quickly switch between the five Silk Central projects that you have most recently accessed.

To open one of the five projects you have most recently opened:

1. In the menu, click **Projects**.
2. In the **Recent Projects** list, click the name of the project that you want to access. You can continue to work with the selected project.

Project Settings

Project settings are settings that are specific to a project, for example the project release date.

Configuring Project Settings

To customize the project settings:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Project Settings** tab to view the current settings. The **Project Settings** page opens.
3. Click **Edit** to modify the current project settings.
4. The **Edit Project Settings** dialog box displays. You can specify the following information:

Build Information File Name Build information files contain project information, including build number, build log location, error log location, and build location. Enter the name of your project's build information file in this field. All test executions will read the build information from this specified file.

Project Release Date Specify the planned release date for your project.

File Extensions to ignore in Results Specify result file types or other file types that should not be saved as results for test executions.

 **Note:** File extensions must be separated by commas, for example, xlg, *_*, res. Changes made in the **Build Information File Name** and **File Extensions to ignore in Results** fields will not effect scheduled tests. To redistribute tasks to execution servers, you must reschedule the tests, or disconnect from and reconnect to the database.

5. Click **OK** to save your project settings.

Filters

Filters provide an efficient means of finding exactly the information you need, while excluding extraneous detail. Filters highlight only those elements that are relevant to your needs, and enable you to quickly sort through requirements, test elements, and execution plans. By defining global filters, you can create complex filter criteria that are available throughout Silk Central without defining filter criteria each time you need to filter a list.

Based on your needs, you can create new filters, edit existing filters, select filters, delete filters, or turn filtering off at the project level. Projects do not contain default filters. You can access and edit filters from the toolbars in the Silk Central units and from the **Project Settings** unit.

 **Note:** Filters are not applied to reports. The **Recent Changes** filter enables you to view project-wide changes and additions that other users have made to tests since your last change acknowledgement. The **Show Changes/Show All** toggle button and the **Acknowledge** button in the **Tests** area help you to find out what changes other users have made. Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Creating Filters

To create a filter:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Filters** tab to view the list of available filters.
3. Click **New Filter**. The **New Filter** dialog box appears.
4. Type a **Name** for the new filter.
This name will be displayed in list boxes when the filter becomes available.
5. Select a **Category** from the list to make the filter available in the **Requirements, Tests, or Execution Planning** area of Silk Central.
6. *Optional:* Type a **Description** for the new filter.
7. *Optional:* Check the **Visible to other users** check box to allow other users to see the filter.
8. *Optional:* Check the **Editable by other users** check box to allow other users to edit the filter.

 **Note:** If **Visible to other users** and **Editable by other users** are checked, the filter is public. To delete non-public (private) filters, you have to be the owner of the filter or you need the **Delete private filters of other users** permission.

9. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.

 **Note:** You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

10. Select a **Property, Operator, and Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|--------------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

11. *Optional:* Click **More** if you want to add more than one filter category to the new filter. Repeat this procedure to define new categories.



Note: If you define more than one filter category, you must define whether the categories need to be fulfilled in addition to the existing categories (AND relationship), or if the filter returns true when one of the filter categories is fulfilled (OR relationship). Select either **AND** or **OR** to define the filter category relationship. You cannot define nested AND, OR relationships.

12. *Optional:* To remove filter categories, click **Fewer**. This removes the last filter category.

13. Click **OK** to save the new filter, or click **Cancel** to abort the operation.

Creating Advanced Filters

Advanced custom filters enable you to combine simple filters to create complex filters that apply multiple filter criteria simultaneously.

To create an advanced custom filter:

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. Click  (**New Filter**) in the toolbar. The **New Filter** dialog box appears.
3. If necessary, click **Advanced** to show the whole dialog box.
4. Click **More** to display a second set of filter-parameter fields with which you can define a second set of filter parameters.
5. Select a logical operator for the application of the filtering queries. For example, with the operator `and` filtered elements must meet both sets of criteria and with the operator `or` filtered elements must meet one, but not both, of the criteria sets.
6. To delete a filter-parameter string, click .
7. To display additional filter-parameter fields and create additional filter queries, click **More**. To remove excess filter-parameter sets, click **Fewer**.

Editing Filters

To edit a filter:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Filters** tab to view the list of available filters.
3. Click the name of the filter you want to edit. The **Edit Filter** dialog box displays.
4. Edit the **Name** and **Description** of the filter.
5. Select a category of filter criteria from the **Selection criteria** list. The available categories depend on the general filter category you have selected.



Note: You can combine filters by selecting **Nested Test Filter** or **Nested Requirement Filter**. Selecting one of these categories allows you to include an existing filter in your new filter.

6. Select a **Property**, **Operator**, and **Value** for the filter from the respective lists.

Property Available properties depend on the filter category that you have selected in the previous step. Defines the property for which you are defining a filter setting. If you have selected an attribute category, the property list includes custom attributes to query against.

Operator Specifies the filter operator. The operator depends on the property type you have selected. For example, if you have selected a property that is based on a string field type, the following operators are available:

| Operator | Description |
|----------|---|
| = | The string equals the defined value. |
| not | The string is different than the defined value. |
| contains | The string contains the defined value. |

| Operator | Description |
|--------------|--|
| not contains | The string does not contain the defined value. |

Value Enter the value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box. Allowed wildcards for strings are * (any amount of characters) and ? (exactly one character).

7. Click **OK** to save the edited filter definition.

Copying Filters

Create a new filter by copying an existing one and changing the settings.

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. In the toolbar of the selected area, select a filter from the list box.
3. Click . The **Edit Filter** dialog box opens with the settings of the original filter.
4. Edit the properties of the filter and click **OK**.

Deleting Filters

To delete a filter:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Filters** tab to view the list of available filters.
3. Click **Delete Filter** for the filter that you want to delete. A confirmation dialog box displays, asking you to confirm the deletion.
4. Click **Yes** to delete the selected filter or **No** to abort the operation. If you click **Yes**, you will be returned to the filters list and the deleted filter is no longer displayed.

Global Filter Details

Global filters allow you to quickly sort through elements in a Silk Central area, highlighting only those that are relevant to your needs.

To view the details of a global filter, click **Project:<Project Name> > Project Settings > Filters** in the menu, and click on the name of the filter.

The specific Silk Central area in which a new filter is available is defined by selecting the general filter category.

Each global filter has to have a property, an operator, and a value from the respective list boxes:

- Property** Available properties depend on the filter category. It defines the property for which you are defining a filter setting. If you selected an attribute category, the property list includes custom attributes to query against.
- Operator** Specifies the filter operator. The operator depends on the property type. For example, if you select a property that is based on a string field type, the available operators are = (equals defined value), `not` (differs from the defined value), `contains` (contains the defined value somewhere in the string), and `does not contain` (does not contain the defined value in the string).
- Value** The value that you want to filter out. Depending on the property type that you have selected, values will either be strings that you can enter into the field, or a selection of predefined values that you can select from the list box.

Recent Changes

The *Recent Changes* filter enables you to efficiently view and acknowledge changes and additions that other users have made to requirements, tests, or execution plans project-wide since your last change acknowledgement.

The two buttons at the far-right of the toolbar, the **Show Changes/Show All** toggle button and the **Acknowledge** button, help you to find out what changes other users have made.

 **Note:** Your system administrator can configure email notifications that alert you to changes that are made to test settings. Email alerts include links that take you directly to a view of recent changes.

Applying Filters

After you have created and stored a custom filter, you can apply that filter to the selected tree. Custom filters can be applied for requirements, tests and execution plans. Only elements that meet applied filter criteria are displayed in the tree.

 **Note:** Filtered requirements are returned in read-only form and can not be edited. The **Edit Properties** button is disabled for filtered requirements.

To apply a stored filter:

1. In the menu, select the appropriate area: **Requirements**, **Tests**, or **Execution Planning**.
2. Select the desired filter from the **Filter** list box on the toolbar.
3. All elements that meet the filter's criteria are displayed.

 **Note:** To remove filtering and display all elements, select **<No Filter>** from the **Filter** list box on the toolbar.

Attributes

You can use attributes to customize the information for tests and execution plans. While some attributes are made available by the integrated functionality of Silk Central, such as priority, components, and platforms, you may want to define custom attributes to categorize tests and execution plans to your needs, or to make tests compatible with specific test cases.

Creating Attributes

To create an attribute:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Attributes** tab to view the list of current attributes.
3. Click **New Attribute**. The **New Attribute** dialog box displays.
4. Type a **Name** for the new attribute. This name will be displayed in list boxes when the attribute becomes available for use.
5. Type a **Description** for the new attribute.
6. Select the attribute **Type**. See the topic *Types for Attributes and Custom Requirements Properties* for descriptions of each type of attribute.
7. Optionally check the **Required (Tests)** check box. Required attributes must be set when creating a new test. Required attributes that are not yet set display with an orange background and a red border on the **New Test** dialog.

 **Note:** If the parent test has a certain attribute already set, the value is automatically filled in for the child test. However, the **Inherit from parent** check box is always unchecked by default. If the parent test does not require a certain attribute, the **Inherit from parent** check box is disabled.

 **Note:** If an attribute is set to **Inactive**, it is not required when creating a new test.

8. Optionally check the **Required (Execution Planning)** check box. Required attributes must be set when creating a new execution plan, configuration suite, or testing cycle. Required attributes that are not yet set display with an orange background and a red border on the creation dialog.

 **Note:** If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.

 **Note:** If an attribute is set to **Inactive**, it is not required when creating a new element.

9. Click **OK**.

Types for Attributes and Custom Requirements Properties

Silk Central supports the following types of attributes and custom requirement properties:

Date The **Date** type supports date values.

List The **List** type supports single select or multi-select lists. It has the following controls:

- **New Item:** Allows you to create entries in the list. Click to show the **New Item** dialog box. Enter the **Name** and **Numeric Weight** for the list item and click **OK**.

 **Note:** The value of the **Numeric Weight** field is used during filtering when the filter criteria uses any of the following operators: **>**, **>=**, **<**, or **<=**. When the filter criteria uses the **=** or **not** operators, the value from the **Name** field is used for comparisons.

- **Order:** Use the fields in this column to sort your attributes.
- **Allow multi-select:** Click this check box to allow users to select more than one values from the list.

 **Note:** You can update an existing single-select list to be a multi-select list. A multi-select list cannot be changed to being a single-select list though.

- **Actions:** Use the icons in this row to delete or to edit each item in the list.

Global List The **Global List** type allows you to reuse items from a centrally managed **List of Values**. A global list lets you maintain a set of items in a single place and use them in multiple projects. These items can be managed (added, removed, or modified) on the **List Of Values Configuration Page**. All changes on the **List Of Values Configuration Page** affect all related **Global List** attributes and custom properties in all projects.

- **Allow multi-select:** Check this check box to allow users to select more than one item from the list.

 **Note:** You can turn a single-select list into a multi-select list. But you can not turn a multi-select list into a single-select list.

 **Note:** When you disable a value from a **List Of Values**, the corresponding **Global List** item remains where it is already used but does not appear any longer in the choice of items in **Edit** operations.

A List of Values can be grouped by another List of Values (the grouping is optional). In such a case, the attributes or custom properties are connected to each other. They are added, edited, and removed coherently. This means ...

- If you add one value, you need to add the other value too.
- If you edit one value, you need to edit the other value too.
- If you remove one value, you need to remove the other value too.

User List The **User List** type supports single-select or multi-select user lists.

Number The **Number** type supports integer or decimal numeric values. It has the following fields:

- **Minimum value:** represents the smallest value that you can enter into the field.
- **Maximum value:** represents the largest value that you can enter into the field.
- **Decimal places:** used to define how many values to the right of the decimal point are valid. Select 0 for an integer value or 1, 2, 3, or 4 for a decimal value.



Note: Attributes and custom requirement properties display their current value if an update is made to their definition (for example, changes to **Minimum value** or **Maximum value**). To apply the new definition, open and save the asset.

Text The **Text** type supports any alphanumeric value. It has the following fields:

- **Maximum length:** represents the maximum number of characters that can be entered into the field. Enter 0 or leave this field empty for unrestricted length (for requirement properties only).

Editing Attributes

To edit an attribute:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Attributes** tab to view the list of current attributes.
3. Click the name of the attribute that you want to edit. The **Edit Attribute** dialog box displays.
4. Edit the **Name** of the attribute.

When the attribute is available for use, the name will display in list boxes for filters, execution plans, and tests. Attributes can be used in global filters for filtering by attributes and can be applied to tests and execution plans.

5. Edit the **Description** of the attribute.
6. See the topic *Types for Attributes and Custom Requirements Properties* for descriptions of each type of attribute. You have different editing options depending on the attribute data type.
7. Optionally check the **Required (Tests)** check box. Required attributes must be set when creating a new test. Required attributes that are not yet set display with an orange background and a red border on the **New Test** dialog.



Note: If the parent test has a certain attribute already set, the value is automatically filled in for the child test. However, the **Inherit from parent** check box is always unchecked by default. If the parent test does not require a certain attribute, the **Inherit from parent** check box is disabled.



Note: If an attribute is set to **Inactive**, it is not required when creating a new test.

8. Optionally check the **Required (Execution Planning)** check box. Required attributes must be set when creating a new execution plan, configuration suite, or testing cycle. Required attributes that are not yet set display with an orange background and a red border on the creation dialog.



Note: If the parent element (for example a configuration suite) has a certain attribute already set, the value is automatically filled in for the child element. However, the **Inherit from parent** check box is always unchecked by default. If the parent element does not require a certain attribute, the **Inherit from parent** check box is disabled.



Note: If an attribute is set to **Inactive**, it is not required when creating a new element.

9. Click **OK**. You are returned to the **Attributes** list.

Deleting Attributes

To delete an attribute:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Attributes** tab to view the list of current attributes.
3. Before you can delete an attribute, you must first deactivate it. In the **Status** column, click the **Active** link or icon and then click **Yes** on the confirmation dialog box to deactivate the attribute.
4. Once the attribute is inactive, click **Delete Attribute** to remove it. A confirmation dialog box displays, asking you to confirm the deletion.
5. Click **Yes** to remove the selected attribute; or click **No** to abort the operation. If you click **Yes** you will be returned to the **Attributes** page, where the removed attribute is no longer displayed.
6. If an error displays, ensure that the selected attribute is not applied to any tests or execution plans, or used in any global filters. You can only delete unused attributes.

Attributes Page

Project:<Project Name> > Project Settings > Attributes

The **Attributes** page lists the attributes that have been created for the current project. For each attribute, the **Attributes** page displays the following columns:

| Column | Description |
|-------------------------------|---|
| Actions | Available actions that can be performed on the attribute. |
| Name | Name of the attribute. When the attribute is available for use, the name will display in list boxes for filters, execution plans, and tests. Attributes can be used in global filters for filtering by attributes and can be applied to tests and execution plans. Click the Edit icon to edit the name of the attribute. |
| Type | Attribute type. See the topic <i>Types for Attributes and Custom Requirements Properties</i> for descriptions of each type of attribute. |
| Status | Status of the attribute, active or inactive. |
| Required (Tests) | Shows whether the attribute is required in the Tests unit. Attributes required for Tests must be set when creating a new test. |
| Required (Execution Planning) | Shows whether the attribute is required in the Execution Planning unit. Attributes required for Execution Planning must be set when creating an execution plan, configuration suite, or testing cycle. |
| Reporting Column | The column name of the attribute in the LQM <i>Reporting</i> table. Use this column name to query the selected attribute within the LQM <i>Reporting</i> table. See the database model documentation for detailed information. |
| Created On | When the attribute was created. |
| Created By | User who created the attribute. |
| Changed On | When the attribute was most recently modified. |
| Changed By | User who most recently modified the attribute. |

Requirement Properties

The following sections describe the two types of requirement properties that are available.

Requirement Properties Page

Project:<Project Name> > Project Settings > Requirement Properties

The **Requirement Properties** page lists the custom requirement properties and the calculated properties that are available for the active project.

Calculated Properties

| Column | Description |
|----------------|---|
| Actions | Available actions that can be performed on the calculated property. |
| Name | Name of the calculated property. Click the Edit icon to edit the calculated property. |
| Formula | Contains the actual formula of the calculated property. |
| Classification | Lists the classifications for the calculated property. |
| Created On | When the property was created. |
| Created By | User who created the property. |
| Changed On | When the property was most recently modified. |
| Changed By | User who most recently modified the property. |

Custom Properties

| Column | Description |
|------------|---|
| Actions | Available actions that can be performed on the custom property. |
| Name | Name of the requirement property. Click the Edit icon to edit the requirement property. |
| Type | Property type. See the <i>Test Attribute and Requirement Property Types</i> topic for descriptions of each type of attribute. |
| Required | Shows whether the property is required. Required custom properties must be set when creating a new requirement. |
| Created On | When the property was created. |
| Created By | User who created the property. |
| Changed On | When the property was most recently modified. |
| Changed By | User who most recently modified the property. |

Calculated Properties

Calculated properties are used to create a single value based on a user-defined formula derived from the values of internal or custom properties. This value is then put into a labelled range called a *classification*. The goal of a calculated property is to be able to view the classification **Name** in grids. The value of a calculated property is shown everywhere where the values of custom properties are shown (Document view, for example). Calculated properties can also be used in filters and quality goals like other internal or custom requirement properties.

Every calculated property has a default classification without a lower bound. All other classifications must have a lower bound. The classification allows you to set a name and range for the value of the calculated property. The name of the classification is how the value will display in the grids. If one or more property values used in the formula are not set, or evaluating the formula on a particular requirement fails (for example, division by zero), --- will appear as the value of the calculated property.

You can use any internal or custom property from within a project in the formula for a calculated property. The internal or custom property must be of type `Number`, `Text`, or `List` with single selection. You can use the following operators in the formula: + - * / ()



Restriction: You can have up to five calculated properties per project. Calculated properties cannot be used within other calculated properties.

Creating a Calculated Property

Calculated properties are used to create a property that is based on a range of values. This procedure will walk you through creating a calculated property named `Custom Risk` with three classifications: `Low`, `Medium`, and `High`. We will use the internal property **Risk** and a custom property named **Impact** of type `Number` (you should create this one in advance).

1. Click **Requirements > Requirement Properties**.
2. Click **New Calculated Property**.
3. In the **Name** field, type `Custom Risk`.
4. In the **Description** field, type `Calculated property using Risk and Impact (L, M, H)`.
5. Select `Business Impact` from the **Insert Property** list. `${Business Impact}` is added to the **Formula** field.
6. Select `*` from the **Insert Operator** list. `${Business Impact}*` is added to the **Formula** field.
7. Select `Business Risk` from the **Insert Property** list. `${Business Impact}*${Business Risk}` is added to the **Formula** field.
8. On the default row of the **Classification** grid, rename the entry to `Low`. Since all calculated properties must contain a default classification without a lower bound, this one will be used.
9. Click **Add Classification**. A new row is added to the **Classification** grid.
10. Name the classification `Medium` and add a **Lower Bound** of 10.
11. Click **Add Classification**. A new row is added to the **Classification** grid.
12. Name the classification `High` and add a **Lower Bound** of 20.
13. Click **OK**.

A calculated property is now created that will be populated with `Low`, `Medium`, or `High` when both the **Impact** and **Risk** properties are set for a requirement.

Calculated Property Page

The **Calculated Property** page is used to add or to edit a calculated property to a project. To access this page click **Requirements > Requirement Properties** and then click an existing calculated property or **New Calculated Property**.

| | |
|-----------------------------|---|
| Name | Contains the name of your calculated property. |
| Description | Enter a meaningful description of your calculated property perhaps referencing the formula or classification. |
| Formula | Contains the formula for your calculated property. You can manually type in the formula or use the Insert Property and Insert Operator lists to do this. If you manually type in the formula, the field names should be contained within the following structure: <code>\${ }</code> . Operators go between this structure. |
| Insert Property | Place your cursor where you want the property to appear in the Formula field and select one of the system or custom properties from the list to add it. |
| Insert Operator | Place your cursor where you want the operator to appear in the Formula field and select an operator. |
| Classifications Grid | The Classifications grid is used to set varying ranges for the values of your calculated properties. Click Add Classification to create a new row in the grid. Use the Name field to identify the classification and the Lower Bound field to determine the lowest value for calculated property. The important thing to know about |

classifications is that the **Name** of the classification is what will appear in grids, not the values of calculated properties. For example if your classification **Name** is *Medium* with a **Lower Bound** of 5 and your calculated property value is 7, as long as your next classification lower bound value is greater than 7, the calculated value will appear as *Medium*.

Internal Property Values

The values of several internal properties can be used in the formulas for calculated properties. The values for supported properties are noted below.

Risk

| Value Name | Internal Value |
|------------|----------------|
| Low | 1 |
| Medium | 2 |
| High | 3 |
| Critical | 4 |

Priority

| Value Name | Internal Value |
|------------|----------------|
| Low | 1 |
| Medium | 2 |
| High | 3 |
| Critical | 4 |

Reviewed

| Value Name | Internal Value |
|------------|----------------|
| No | 0 |
| Yes | 1 |

Custom Requirement Properties

You can add custom property fields across all requirements in the selected project in **Requirements > Requirement Properties**. Custom properties can subsequently be edited alongside the default properties on the **Edit Requirements** dialog box. Custom properties are displayed in **Requirements > Details View > Properties**. If the only value of a property is a URL, Silk Central automatically displays it as a link, which enables direct access to other web applications from within Silk Central.

Creating a Custom Requirement Property

To create a new custom requirement property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirement Properties** tab.
3. Click **New Requirement Property** to display the **New Requirement Property** dialog box.
4. Type a **Name** for the new property and optionally a **Description**.

5. Select the data **Type** of the new property from the **Type** list. See the *Custom Attribute and Requirement Property Types* topic for descriptions of each type of attribute.
6. Optionally check the **Required** check box. Required custom properties must be set when creating a new requirement. On the **New Requirement** dialog, required properties that are not yet set display with an orange background and a red border.



Note: Existing requirements are not affected. But when you edit existing requirements, newly created required custom properties must be set as well to be able to save the requirement.



Note: Child requirements do not inherit required properties. If desired, these must be set explicitly for child requirements as well.

7. Click **OK**.

Your custom property is available to all requirements in the active project.

Types for Attributes and Custom Requirements Properties

Silk Central supports the following types of attributes and custom requirement properties:

Date

The **Date** type supports date values.

List

The **List** type supports single select or multi-select lists. It has the following controls:

- **New Item:** Allows you to create entries in the list. Click to show the **New Item** dialog box. Enter the **Name** and **Numeric Weight** for the list item and click **OK**.



Note: The value of the **Numeric Weight** field is used during filtering when the filter criteria uses any of the following operators: **>**, **>=**, **<**, or **<=**. When the filter criteria uses the **=** or **not** operators, the value from the **Name** field is used for comparisons.

- **Order:** Use the fields in this column to sort your attributes.
- **Allow multi-select:** Click this check box to allow users to select more than one values from the list.



Note: You can update an existing single-select list to be a multi-select list. A multi-select list cannot be changed to being a single-select list though.

- **Actions:** Use the icons in this row to delete or to edit each item in the list.

Global List

The **Global List** type allows you to reuse items from a centrally managed **List of Values**. A global list lets you maintain a set of items in a single place and use them in multiple projects. These items can be managed (added, removed, or modified) on the **List Of Values Configuration Page**. All changes on the **List Of Values Configuration Page** affect all related **Global List** attributes and custom properties in all projects.

- **Allow multi-select:** Check this check box to allow users to select more than one item from the list.



Note: You can turn a single-select list into a multi-select list. But you can not turn a multi-select list into a single-select list.



Note: When you disable a value from a **List Of Values**, the corresponding **Global List** item remains where it is already used but does not appear any longer in the choice of items in **Edit** operations.

A List of Values can be grouped by another List of Values (the grouping is optional). In such a case, the attributes or custom properties are connected to each other. They are added, edited, and removed coherently. This means ...

- If you add one value, you need to add the other value too.
- If you edit one value, you need to edit the other value too.

- If you remove one value, you need to remove the other value too.

User List The **User List** type supports single-select or multi-select user lists.

Number The **Number** type supports integer or decimal numeric values. It has the following fields:

- **Minimum value:** represents the smallest value that you can enter into the field.
- **Maximum value:** represents the largest value that you can enter into the field.
- **Decimal places:** used to define how many values to the right of the decimal point are valid. Select 0 for an integer value or 1, 2, 3, or 4 for a decimal value.



Note: Attributes and custom requirement properties display their current value if an update is made to their definition (for example, changes to **Minimum value** or **Maximum value**). To apply the new definition, open and save the asset.

Text The **Text** type supports any alphanumeric value. It has the following fields:

- **Maximum length:** represents the maximum number of characters that can be entered into the field. Enter 0 or leave this field empty for unrestricted length (for requirement properties only).

Editing Custom Requirement Properties

To edit a previously created custom requirement property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirement Properties** tab.
3. Click the name of the property you want to edit. The **Edit Requirement Property** dialog box displays.
4. Edit the name of the property in the **Name** field.

When the property is available for use, the name will display in lists for filters and requirements. Properties can be used in global filters for filtering by requirement properties and can be applied to requirements.

5. Edit the **Description** of the property.
6. See the *Custom Attribute and Requirement Property Types* topic for descriptions of each type of requirement property. You have different editing options depending on the property data type.
7. Optionally check the **Required** check box. Required custom properties must be set when creating a new requirement. On the **New Requirement** dialog, required properties that are not yet set display with an orange background and a red border.



Note: Existing requirements are not affected. But when you edit existing requirements, newly created required custom properties must be set as well to be able to save the requirement.



Note: Child requirements do not inherit required properties. If desired, these must be set explicitly for child requirements as well.

8. Click **OK** to save your changes.

Deleting Custom Requirement Properties

To delete a previously created custom requirement property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirement Properties** tab.
3. Click **X** . A confirmation dialog box appears.
4. Click **Yes** to confirm the deletion.

Step Properties

Use step properties to extend a manual test step with additional properties. Add a custom step property to a project in **Projects:<Project Name> > Project Settings** to add the property to all manual test steps in the project. Add a custom step property to a library in the **Libraries** tree to add a custom step property to all manual test steps in the library. Step properties can subsequently be edited alongside the default properties on the **Edit Manual Test Step** dialog box.

Creating Step Properties

To create a new step property:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Step Properties** tab.
3. Click **New Property**. The **New Step Property** dialog box opens.
4. Enter a name for the new property in the **Name** field.



Note: Step property fields are always declared as type string.

5. Click **OK** to make your property available to all manual test steps in the selected Silk Central project.



Note: To create a step property for a library, select the library node in the **Libraries** tree and continue with the second step.

Editing Step Properties

To edit a previously created step property

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Step Properties** tab.
3. Click on the name of the property that you want to edit. The **Edit Step Property** dialog opens.
4. Edit the name of the property in the **Name** field.
5. Click **OK** to save your changes, or click **Cancel** to abort the operation without saving.

Deleting Step Properties

To delete a previously created step property

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Step Properties** tab.
3. Click **Delete Property** in the **Actions** column of the property that you want to delete. A confirmation dialog box displays, asking you to confirm the deletion.
4. Click **Yes** to complete the operation, or **No** to abort.

Step Properties Page

Project:<Project Name> > Project Settings > Step Properties

The **Step Properties** page lists all properties that can be populated into manual test steps across the active project. For each custom step property, the **Step Properties** page displays the following columns:

| Column | Description |
|---------|---|
| Name | Name of the step property. |
| Actions | The following actions can be performed on a property: |

| Column | Description |
|--------|--|
| | <ul style="list-style-type: none"> • Delete Property • Move Up • Move Down |

Change Notification

Silk Central can notify you by email when requirements or tests are changed by other users. Each user has the option of activating change-notification. Once notification has been enabled, perform an initial acknowledgment. From then on, you can view and acknowledge changes that have occurred since your last acknowledgment. To avoid numerous notifications, only a single email alert is sent to you when a change is made, regardless of how many changes other users may have made since your last acknowledgment. Email alerts include links that take you directly to a view of recent changes.

Before you can activate change notification for requirements or tests, you must configure your email address in the user settings of Silk Central.



Note: Change notification only works if an email server has been configured by your administrator. If change notification has not been enabled, please contact your Silk Central administrator.

Enabling Change Notification

To enable change notification:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Notifications** tab.
3. Click **Configure Email Notification** to open the **Configure Email Notification** dialog box.
4. If you want to be notified by email when changes are made to requirements in the currently selected project, check the **Changes on Requirements** check box.
5. If you want to be notified by email when changes are made to tests within the currently selected project, check the **Changes on Tests** check box.
6. Click **OK** to save the notification settings, or click **Cancel** to abort the operation without saving changes.

You will be notified by email about changes for which you have activated notification.

Disabling Change Notification

To disable change notification:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Notifications** tab.
3. Click **Configure Email Notification** to open the **Configure Email Notification** dialog box.
4. If you do not want to be notified by email when changes are made to requirements in the currently selected project, uncheck the **Changes on Requirements** check box.
5. If you do not want to be notified by email when changes are made to tests in the currently selected project, uncheck the **Changes on Tests** check box.
6. Click **OK** to save the notification settings or click **Cancel** to abort the operation without saving changes.

Changes Triggering Change Notification

When you have activated change notification to inform you of changes that are made to requirements or tests an email alert is sent to you, following your logout. The email alert is sent to you if one or more of the following settings are changed:

| Area | Changes |
|---------------------|--|
| Requirements | <ul style="list-style-type: none"> • A requirement is created or deleted. • The name or description of a requirement is edited. • A system property is edited. • A requirement is set as obsolete. • A requirement is recovered. • A test is assigned to or removed from a requirement. • A custom property of a requirement is created, edited, or deleted. |
| Tests | <ul style="list-style-type: none"> • A container is created or edited. • A product is edited. • A source control profile is edited. • "Clear working folder" is edited. • The root node is edited. • The custom data directory is edited. • The include directory is edited. • The hidden test properties are edited. • The Silk Test Classic interface is edited. • A test folder is created, edited, or deleted. • A test is created, edited, or deleted. • The planned time is edited. • A test step is added or edited. |

Notifications Page

Project:<Project Name> > Project Settings > Notifications

The **Notifications** page lists the notification events for the active project. For each notification event, the **Notifications** page displays the following columns:

| Column | Description |
|---------------------|---|
| Notification Events | Name of the notification event that has been set up for the active project. |
| Status | Status of the notification event. Active or inactive. |

Click the **Configure...** button in each section to edit the values.

Email

When an email notification event is activated, a notification email is sent to the user that activated the event, the first time one of the specified settings is changed.



Note: You must have specified an email address to be able to receive email notifications.

Requirement Flagging

You can configure notifications differently for internal and external requirements. All requirement flagging is enabled by default. For more information, see the *Flags* topic.

Requirements Management Tools Integration



Important: Before upgrading from Silk Central 15.0 or earlier to the current version, perform a **Synchronize All** to synchronize all your requirements with your integrated requirements management tools.

Integrating external requirements management enables you to coordinate the requirements management features of Silk Central with other tools you may already be working with. Integration is configured on the **Requirements Management** page in **Project:<Project Name> > Project Settings**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

Silk Central supports integration with external requirements management systems (RMS) through its open interface. Creating a plug-in and integrating it into Silk Central allows integrating any RMS. Refer to the *Silk Central API Help* for information about the interfaces that enable proper integration of external RMS.

Synchronizing Requirements

Enabling synchronization of requirements between Silk Central and an external RMS enables Silk Central to receive changes that occur in the external RMS whenever a synchronization is executed. If a project has external RMS integration enabled, the master system for requirements is automatically the external system. This means that synchronization is always from the external RMS tool to Silk Central. Requirements can no longer be edited in Silk Central.

Property mapping functionality allows you to map property fields between Silk Central and external requirement tools. For example, a custom field in Silk Central called `user` might be equivalent to a property field in Caliber called `Field_2`. The property mapping feature ensures that changes to requirement-property fields are accurately refreshed between projects. If you don't use the property mapping feature, only the name and the description of the requirements are mapped. For more information, see *Editing Property Mapping*.

Requirements can be synchronized in one of several ways:

Manual synchronization

Click **Synchronize Changes** at the root folder level on the **Properties** page to synchronize only requirements that have changed since the last synchronization.

Forced manual synchronization

Click **Synchronize All** at the root folder level on the **Properties** page to force synchronization of all requirements.

Automatic scheduled synchronization

Based on globally defined Silk Central schedules.

Automatic online synchronization

Changes to requirements are automatically propagated between tools. This option is available for Caliber only. It requires Caliber client installation on the application server and MPX enabled on the Caliber Server (refer to the Caliber Installation Guide for detailed information on enabling MPX). To enable automatic synchronization between Caliber and Silk Central, add the file `ss.jar` to the application server at `Program Files\Silk\Silk Central <version>\lib` and restart the application server. Contact customer care to obtain the file. Requirement data is automatically updated in Silk Central when changes are made in Caliber and traces in Caliber are updated when test assignment changes are performed in Silk Central. This type of online synchronization is only available when projects are configured with the current baseline.

Automatic synchronization of requirements between Silk Central and external requirements management tools can be configured to occur based on global schedules. For details on configuring global schedules, see the *Administration* topics in this Help.



Note: The **Open Caliber** buttons open whatever program is registered as the default program for opening files of extension `.crm`. On some machines, this may be the requirement viewer, rather than Caliber. This behavior can be changed by your administrator. The client program is called `caliberrm.exe`. When properly configured, the program opens to the requirement that is selected in Silk Central.

The binder icon on the project node of the **Requirements** tree indicates the status of RM integration for the project:

No configuration RM integration is not available.

Manual configuration Requirement import, upload, and synchronization can be done only by clicking the corresponding buttons on the project node in **Requirements** >  > **Properties**.

At the project level, the **Properties** page includes the following properties:

| Property | Description |
|-----------------------------|---|
| System | The external tool with which integration has been enabled. |
| Status | Whether or not integration has been enabled. |
| Project Name | The name of the external project that the Silk Central project is associated with. |
| Requirement Types | The requirement types that are shared between projects. |
| Last Synchronization | Date and time of the last synchronization. |
| Last Synchronization Status | The status of the last synchronization including the number of created, updated, and deleted items. |



Note: When integration between Caliber and Silk Central with automatic online synchronization has been enabled, the project node displays the current status of the online requirements change listener. The three possible statuses for such projects are: `Connected` (synchronized), `Reconnected` (synchronization recommended), and `Disconnected`.

Synchronizing Requirements Across Tools



Note: Use the property mapping feature to map property fields. If you don't use the property mapping feature, only the name and the description of the requirements are mapped. For more information, see *Editing Property Mapping*.

To synchronize requirements between Silk Central and an externally configured requirements management tool

1. In the menu, click **Requirements** > **Details View**.
2. Select the Project node in the **Requirements** tree.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **Synchronize Changes**.
5. Click **Yes** on the **Synchronize Requirements** confirmation dialog box to begin synchronization. A dialog box opens when synchronization is complete, displaying synchronization statistics, including the number of requirements that have been created, updated, and deleted.
6. Click **OK** to complete the synchronization. Any updates that were made to mapped requirements in your externally configured requirements management tool are now reflected in the **Requirements** tree.

Synchronizing Requirements based on Schedules

You can configure global schedules to automatically synchronize requirements between Silk Central and an external requirements management tool. To enable automatic synchronization between Caliber and Silk Central, add the file `ss.jar` to the application server at `Program Files\Silk\Silk Central <version>\lib` and restart the application server. To obtain the file, contact customer care.

To synchronize requirements based on globally defined schedules:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirements Management** tab.
3. Click **Edit Schedules**. The **Edit Schedules** dialog box opens.
4. Select a predefined global schedule from the **Synchronize changes** list to define when changes in the external system should be synchronized with Silk Central.
5. Select a predefined global schedule from the **Synchronize all** list to define when all requirements in the external system should be synchronized with Silk Central.
6. Click **OK**.

For information on creating global schedules, see [Adding Global Schedules](#).

Defining Email Notification for Automatic Synchronization Events

You can define an email notification to alert users when errors occur during automated synchronization of requirements between Silk Central and external requirements management tools. All notification recipients receive copies of the synchronization log files.

To define an email notification for automatic synchronization events:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Requirements Management** tab.
3. Click **Edit Notification**. The **Edit Notification** dialog box displays.
4. Check the **Enable notification** check box.
5. Select a user name from the **Username** list box.
6. If required, add additional email addresses for other recipients in the **Other email addresses** field. Use semicolons to separate multiple email addresses.
7. Click **OK**.

Atlassian Jira Software Integration

Silk Central offers a requirements integration with Atlassian Jira Software, which is based on the project and issue tracking system Jira. All issue types (the generic object type in Jira) can be synchronized. Since Atlassian Jira Software offers the creation of custom issue types, you can specify a list of Atlassian Jira Software issue types in Silk Central that you want to synchronize, such as for example `Bug` or `Improvement`.

Atlassian Jira Software versions that are currently supported for integration with Silk Central:

- Atlassian Jira Software 6, 7, 8. Versions 6 and 7 are supported but not tested.
- Atlassian Jira Software Cloud. Not supported are *next-gen* projects and projects residing in a cloud instance containing *next-gen* projects.

Limitations: Synchronizing Jira requirements only works for requirements of the configured Jira project. If requirements are linked to requirements (epics) which are located in a different Jira project, these requirements will not be synchronized.

Enabling Integration with Atlassian Jira Software

To enable integration with Atlassian Jira Software:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.

5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **Atlassian Jira Software** from the list and click **Next**. The **Edit Configuration** dialog box opens.

7. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.

8. Enter the **URL** of your Atlassian Jira Software server.

9. Type a valid **Username** and **Password**. The user name needs to be a valid Jira user profile name.

These credentials will be used to access your Jira issue tracking system.



Important: Use the credentials of a Jira account with sufficient permissions, as missing permissions in Jira can cause inconsistencies when the plug-in attempts to add or update issues through this account. The Jira account requires the following permissions for each project that is used in Silk Central:

- Browse Projects
- Create Issues
- Assignable User
- Modify Reporter

Additionally, the Jira account needs to be able to access (log in to) your Jira application.

Use the following credentials to integrate with Jira cloud:

- **Username:** A Jira user profile name.
- **Password:** A Jira API token. For more information on how to generate the API token, refer to the Jira documentation.

Use the following credentials to integrate with an on-premise Jira instance:

- **Username:** A Jira user profile name.
- **Password:** A Jira password.

10. In the **Issue Types** field, specify the issue types that you want to synchronize, separated by semicolons (;).

All issue types (the generic object type in Jira) can be synchronized. Since Atlassian Jira Software offers the creation of custom issue types, you can specify a list of Atlassian Jira Software issue types in Silk Central that you want to synchronize, such as for example `Bug` or `Improvement`.

11. *Optional:* To limit the amount of requirements that are being synchronized, you can use the **Custom JQL** field to pre-filter requirements before they are being synchronized. Create a query using Jira Query Language (JQL) within your Jira system, then copy and paste the string into the **Custom JQL** field. For more information on JQL, visit Atlassian's [Advanced Searching Documentation](#).



Tip:

- Do not include `ORDER BY` statements in your JQL query.
- The JQL string may not exceed 2000 characters.
- The JQL query is used additionally to your configured project and issue types, it does not overrule those settings.

12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.

A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

13. Click **Load**. The **Project** list is populated with all the projects in Jira that you have permissions for.
14. From the **Project** list, select the external project with which the Silk Central project is to be integrated.
15. Click **OK** to save your settings.

Atlassian Jira Software Property Mappings

The following properties have a default mapping between Jira and Silk Central:

- User stories that are in Jira's `Backlog` are reflected in Silk Central by having the `Sprint` property value set to `Backlog`. This enables you to easily search for them by creating a filter.
- The default `Priority` values of Jira's user stories are mapped to Silk Central's `Priority` property. If you are not using Jira's default `Priority` values, you can customize this mapping.

Viewing Jira Backlog Stories in Silk Central

1. In the menu, click **Requirements > Details View**.
2. Create a new filter with the following criteria:
 - **Selection Criteria:** `External Custom Attributes`
 - **Property:** `Sprint`
 - **Value:** `Backlog`
3. Click **Apply** to apply the filter immediately or **Save and apply** to save the filter for future reuse.

Mapping Jira Priority values to Silk Central's Priority

The default `Priority` values of Jira's user stories are mapped to Silk Central's `Priority` property as follows:

- `Blocker=SC_Priority_Critical`
- `Critical=SC_Priority_High`
- `Major=SC_Priority_Medium`
- `Minor=SC_Priority_Low`
- `Trivial=SC_Priority_Low`

If you are not using Jira's default `Priority` values, or if you want to customize the mapping, your Silk Central administrator should customize the settings as follows:

1. Stop the application server.
2. Open the file `propertyMappings.properties` with a text editor. This file is located in the Jira plugin-file `JIRA Agile.zip` on the application server in the `plugins` folder.
3. Search for `SC_Priority=Priority`, beneath which you can customize the mapping of Jira's `Priority` values to Silk Central's `Priority`.



Note: Silk Central `Priority` values require the prefix `SC_Priority_`.

4. Save and close the file
5. Re-start the application server.

CA Agile Central Integration

Integrating the project management tool CA Agile Central enables you to define user stories which you can then use as requirements in Silk Central. You can then create tests in Silk Central to cover these requirements. To update the tests you have created with any changes to the corresponding user stories in CA Agile Central, synchronize the requirements.

The following user story properties are always synchronized from the CA Agile Central REST Service. Custom fields are also synchronized if they are specified in the integration configuration. You can use these properties to help filter and categorize the user stories.:

| Property | Description |
|-------------------------------------|--|
| Name | The name of the user story in CA Agile Central. |
| CA Agile Central ID | The identifier of the user story in CA Agile Central. This property is the External ID of the requirement in Silk Central. |
| Description | The description of the user story in CA Agile Central. |
| Iteration | The iteration in CA Agile Central in which the user story is included. |
| Project | The project in CA Agile Central in which the user story is included. |
| Release | The project release in which the user story is included. |
| State | The state of progress of the user story in CA Agile Central. The following states are available: <ul style="list-style-type: none"> • Defined • In-Progress • Completed • Accepted |
| Plan Estimate (Story Points) | The estimated time in CA Agile Central for the user story to be completed. The time is estimated in story points. |

Enabling Integration with CA Agile Central

To enable integration with CA Agile Central:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **CA Agile Central** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Type the **URL**.
9. Type valid **Username** and **Password** credentials for CA Agile Central, or type a valid **API Key** instead. An API Key can be used to access your CA Agile Central subscription data without using your username and password. For more information about CA Agile Central API Keys, refer to the [CA Agile Central Application Manager page](#).
10. *Optional:* If direct access to the Internet is restricted, and the CA Agile Central REST service is located in a machine outside the scope of the internal network, use the **Proxy Host** and **Proxy Port** fields to specify a proxy through which Silk Central can connect to CA Agile Central.



Note: To connect to CA Agile Central by using a proxy, you need to fill out both fields, **Proxy Host** and **Proxy Port**.

11. *Optional:* If you want to synchronize custom fields, type their names into the **Custom Fields** field, separated by commas. The internal field names must be used, not the display names.

12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct. A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

13. Click **OK**.
14. Click **Load**. The **Project** list box is populated with all the projects from all workspaces in CA Agile Central that you have permissions for.
15. From the **Project** list box, select the external project with which the Silk Central project is to be integrated.
16. Click **OK** to save your settings.

Caliber Integration

This section describes how to integrate Caliber with Silk Central.

Configuring integration with Caliber requires the definition of Caliber login credentials. Whenever requirements are synchronized between Silk Central and Caliber, these credentials are used to login to Caliber, thus checking out a Caliber license. The license is set free as soon as the synchronization process has completed. We recommend creating a dedicated Caliber user for synchronization purposes, which should be used by all Silk Central requirements integrations with Caliber. This ensures that only a single Caliber license is used for the process of synchronization.

Caliber versions that are currently supported for integration with Silk Central: Caliber 11.4, 11.5



Note: Caliber user defined attribute **Multiple selection user list** and **Multiple selection group list** must be mapped to the Silk Central text attribute type and not list type.

Enabling Integration with Caliber

To enable integration with Caliber:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **Caliber** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Type the **Hostname** of the machine where the external server is installed.
9. Type valid **Username** and **Password** credentials for the requirements management server.
10. Click **Test Connection** to confirm that the host and user credentials you have entered are correct. A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

11. Click **OK**.
12. From the **Project** list box, select the external project with which the Silk Central project is to be integrated. The requirement types that are available with the selected project are automatically

populated into the **Requirement Types** field. The baselines that are available with the selected project are automatically populated into the **Baseline** field.

13. Select a **Baseline** from the external project that should be integrated with the Silk Central project.

Your selections are displayed on the **Edit Configuration** dialog box.

14. Click **OK**.

15. Select one or more requirement types from the external project that should be integrated with the Silk Central project.

Use **Ctrl+Click** to select multiple requirement types.

16. Click **OK** to save your settings.

Baseline Support for Caliber Integration

You can select the current baseline or existing user-defined baselines for Caliber integration. You cannot import modified baselined requirements into Silk Central. Requirements that are not of the current baseline can only be changed in Caliber if the version of the requirement that is used for the baseline is changed. Such changes are only updated within Silk Central requirements when a manual or scheduled synchronization is performed.

You can change a baseline after you import it into Silk Central. You can change the configured baseline to a different user-defined baseline or the current baseline. After such a change, the next synchronization of the baseline, either manual or scheduled, will update the Silk Central project and update, create, or delete requirements as required. When a baseline is changed, a message displays that states that the changes will take effect after the next synchronization.

Handling Test Assignments in Caliber

Assigned tests are displayed, managed, and created as traces ("Trace to") of synchronized requirements in Caliber.

The External Traceability for Silk Central must be enabled for a Caliber project within the Caliber Administrator and the correct Silk Central front-end server must be configured. Click **Edit** for the Silk Central External Traceability for correct project within the Caliber Administrator.

When a project is synchronized with a Silk Central project, the assigned tests of the synchronized Silk Central requirements will display as traces ("Trace To") of the Caliber requirement. If you edit these assignments in Silk Central, the changes will be reflected immediately in Caliber.

Copying Caliber-Integrated Projects

To manage Caliber baselines when copying Silk Central projects:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Requirements Management** tab.
3. Verify that the baseline you want to save is selected.



Note: If a baseline is changed, you must perform a synchronization to update the project requirements with the baseline changes, before you can copy an associated Silk Central project. The integration configuration is only copied if you select a baseline other than the current baseline. If you select the current baseline, you need to specify if you want to keep the integration configuration in the original project or move it to the copied project.

4. If the baseline that you want to save is not selected, click **Edit Configuration**. The **Edit Configuration** dialog box displays.
5. Click **Browse** next to the **Project name** field. The **Browse Projects** dialog box opens.
6. Select the baseline you want to save, then confirm your selection.
7. In the menu, click **Projects > Project List**. The **Projects** page displays, listing all existing projects and project baselines.

8. Click  in the **Actions** column of the project you want to copy.

 **Note:** For full details on copying a project, see the *Administration* topics in this Help.

The **Copy Project** dialog box displays.

9. Select the items you want to copy into the new project, then confirm your selection.

10. Apply the baseline that you want to continue working with to the Silk Central project.

 **Note:** After copying a project, the original project and the copy are identical. Define on which you will continue working on by applying the corresponding baseline.

Creating a Baseline for a Caliber-Integrated Project

To create a baseline of a Caliber-integrated Silk Central project:

1. Create a baseline of the Silk Central project. A Silk Central message box displays during the process, asking if you want to enable the Caliber integration for the new baseline.

2. Click **Yes**. When the new baseline is created, the **Baseline Project - Adapt Project Settings** dialog box opens.

3. Click **Edit** to change the requirements integration settings.

The **Edit Configuration** dialog box opens.

4. Click **Browse** next to the **Project name** field. The **Browse Projects** dialog box opens.

5. Select a Caliber project, a baseline, and the requirement type for synchronization.

6. Click **Ok** to close the **Browse Projects** dialog box.

7. Click **Ok** to close the **Edit Configuration** dialog box.

8. Click **Finish** to close the **Baseline Project - Adapt Project Settings** dialog box.

Generating Tests from Scenarios and Simulations

Generate tests from the Caliber objects **Scenario** and **Simulation**.

To generate tests from the Caliber objects **Scenario** and **Simulation**:

1. Establish integration with Caliber.

For additional information, see *Enabling Integration with Caliber*.

2. Select the **Scenario** or **Simulation** in the **Requirements** tree from which you wish to generate tests.

3. On the **Properties** tab, click **Generate Tests**. The **Generate Tests** dialog box displays.

4. Use **Ctrl+Click** or **Shift+Click** to select the tests that you want to generate.

5. Click **Generate All** to generate all tests, or click **Generate Selected** to generate the selected tests. The **Tests** tree displays.

6. Select the folder to which you want to add the generated tests.

 **Note:** Tests that already exist in the selected destination folder are not created, but updated. No tests in the folder are deleted.

7. Click **OK**.

IBM Rational DOORS Integration

This section describes how to integrate Silk Central and IBM Rational DOORS. Silk Central 20.6 offers out-of-the-box support for IBM Rational DOORS, without the need to customize anything in your DOORS installation. If you are upgrading from Silk Central 15.5 or earlier though, and using customized dxl files, you will need to overwrite the dxl files in the `lib\dxl` folder within your Silk Central installation folder with your customized dxl files.

IBM Rational DOORS versions that are currently supported for integration with Silk Central: IBM Rational DOORS 9.5, 9.6

Enabling Integration with IBM Rational DOORS

To integrate Silk Central and DOORS, install the DOORS client on the Silk Central front-end server machine. If you use more than one front-end server machines, you must install the DOORS client on each front-end server that you have configured for integration with DOORS.

To enable integration with DOORS:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **IBM Rational DOORS** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.

8. In the **RM service URL** field, type the URL of the Silk Central DOORS requirement Web Service.

The default value should point to the correct location already. For example `http://MySCTMHost:19120/services/doorsrequirementsmanagement`.

9. Type valid **Username** and **Password** credentials for the requirements management server.
10. The default DOORS client installation path is displayed in the **DOORS Installation Path** field. If this path is not correct, enter the correct destination within the front-end server directory structure.
11. *Optional:* To synchronize only specific requirements, based on a filter that has been specified and activated for a view in DOORS, specify the name of the view in the **DOORS Filter View** field.
12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.

A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: If **Test Connection** fails with the error 80004005 Unspecified error: Access denied: Doors session is not authenticated, the issue might be that the DOORS setting **Prompt to open all modules from last session on startup** is enabled. To solve this issue, disable this setting in the DOORS options. Consult your system administrator if you are not able to establish a connection.

13. Click **Browse** next to the **Project name** field to open the **Browse Requirement Types** dialog box. From the **Project** field, select the external project with which the Silk Central project is to be synchronized.

The requirement types that are available with the selected project are automatically populated into the **Requirement types** field. Select the requirement types that are to be synchronized and click **OK**. Use **Ctrl+Click** to select multiple requirement types.

Your selections are now displayed on the **Edit Configuration** dialog box.

14. Click **OK** to save your settings.



Caution: As the DOORS application object is used for communication, and this object does not support login data, but rather requires a running DOORS client, Silk Central starts each DOORS client process with the provided login data and then uses that same data for all subsequent application objects. Therefore only one set of DOORS login credentials is supported for communication at one time. It is recommended that you use the same DOORS credentials for all configurations so that integration tasks can be performed on the front-end server for all projects at

the same time. When a second set of credentials is used, the second set only works after all sessions using of the first set of credentials have timed out.

IBM Rational DOORS Next Generation Integration

This section describes how to integrate Silk Central and IBM Rational DOORS Next Generation. Silk Central 20.6 offers out-of-the-box support for IBM Rational DOORS Next Generation, without the need to customize anything in your DOORS installation.



Note: All artifacts of the project are synchronized, except the DOORS collection artifacts **Module** and **Collection** themselves.

IBM Rational DOORS Next Generation versions that are currently supported for integration with Silk Central: IBM Rational DOORS Next Generation 6.0

Enabling Integration with IBM Rational DOORS Next Generation

To enable integration with IBM Rational DOORS Next Generation:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **IBM Rational DOORS Next Generation** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. In the **DOORS Next Generation URL** field, type the fully qualified URL of the IBM Rational DOORS Next Generation server, for example `https://MyDOORSHost.MyDomain.com:9443/rm`.
9. In the **Authentication URL** field, type the fully qualified URL of the IBM Rational Jazz Team Server, for example `https://MyDOORSHost.MyDomain.com:9443/jts`.



Note: The new IBM Rational Authentication Server is not supported for authentication, only the IBM Rational Jazz Team Server.

10. Type valid **Username** and **Password** credentials for the IBM Rational DOORS Next Generation server.
11. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.
A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

12. Click **Load** next to the **Project name** list to populate the list with the available DOORS projects. Select the external project with which the Silk Central project is to be synchronized.
13. Click **OK** to save your settings.

SAP Solution Manager Integration

To purchase the SAP Solution Manager Plug-In, please contact your sales representative.

SAP Solution Manager facilitates technical support for distributed systems, covering solution deployment, operation, and continuous improvement. SAP Solution Manager is a centralized, robust application

management and administration solution, that combines tools, content, and direct access to SAP to increase the reliability of solutions and lower total cost of ownership. For additional information on the SAP Solution Manager, refer to the SAP Solution Manager documentation.

The integration between Silk Central and SAP Solution Manager allows for the synchronization of SAP Solution Manager business processes and incidents as Silk Central requirements and issues.

System Requirements for SAP Solution Manager Integration

The following are the system requirements for the SAP Solution Manager integration plugin:

- SAP Solution Manager 7.2.
- Silk Central 19.0 or later.
- The *Silk Central Integration Hub for SAP Solution Manager* must be installed in SAP Solution Manager.

Configuring SAP Solution Manager

Configure SAP Solution Manager to enable the integration with Silk Central.

For additional information, see

[Configuration_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#) and [User_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#), which are included in the SAP Solution Manager plug-in.

For this integration to get valid links from SAP Solution Manager to Silk Central, the front-end server machine of Silk Central must have a valid fully qualified name.

Adding SAP Solution Manager Issue Tracking Profiles

To add a SAP Solution Manager issue tracking profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Issue Tracking** tab. The **Issue Tracking** page opens, listing all of the issue tracking profiles that have been created for the system.
3. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Type a **Description** for the new profile.
6. Select **SAP Solution Manager** from the **Type** list box.
7. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
8. Type the **URL** of the **Z-SCTM_ADAPTER** web service of SAP Solution Manager.
9. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
10. Click **Load Default Incident Type** to load all possible incident types and populate the **Default Incident Type** list box.
11. Select an incident type from the **Default Incident Type** list box. If you are creating links to existing incidents in SAP Solution Manager and you do not specify the incident type, this type will be used for it.
12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

13. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.

- Click **Yes** to proceed with the related *Mapping Issue States* procedure.
- Click **No** to map issue states later.

Enabling Requirements Integration with SAP Solution Manager

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **SAP Solution Manager** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Enter the **URL** of the `Z_SCTM_ADAPTER` web service of SAP Solution Manager.
9. Enter the **Username** and **Password**.
10. Click **Test Connection** to confirm that the host and user credentials are correct. If the settings are correct, a **Test connection was successful** message displays.
11. Click **Load** to load all available solution and branch combinations and select the correct one.
Silk Central stores such combinations of solutions and branches as projects.
12. *Optional:* Check the **Enable creation of unassigned requirements** check box to enable creation and editing of unmapped requirements in Silk Central projects that are configured for integration.
13. Click **OK** to confirm and close the dialog.

Working with External Properties

This section explains how to work with external properties in Silk Central.

Editing External Properties

To edit external properties:

1. In the menu, click **Requirements > Details View**.
2. Select the requirement for which you intend to edit external properties.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **Edit External Properties**.

The **Edit External Properties** dialog box displays. All properties of the external requirement are displayed here. Edit all properties as required.



Note: Editable properties on this dialog box offer input fields and controls with which you can edit the properties. If a mapping rule exists for an attribute, the attribute will be tagged with a trailing asterisk (*).

5. Click **OK** to save your changes and close the dialog box.

Viewing External Properties

To view external properties:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **View External Properties**. The **View External Properties** dialog box displays. All properties of the external requirement are displayed here.
5. Close the dialog box.

Editing Property Mapping

The property mapping functionality allows you to map property fields between Silk Central and external requirements management tools. For example, a custom requirement property in Silk Central called `User` might be equivalent to a custom property in Caliber called `User_ID`. The property mapping feature ensures that requirement property fields are accurately populated between projects during requirement uploading and importing. If there are multiple requirement types, you must map each type separately.



Note: If you don't use the property mapping feature, only the name and the description of the requirements are mapped.

To edit property mapping:



Note: The following applies for synchronizing properties with boolean values: since Silk Central does not support the boolean data type, you need to create a single select list with the following two list items:

- Name = `Yes`, Numeric Weight = 1
- Name = `No`, Numeric Weight = 0



Note: When you map lists, requirement properties from both products need to have the same named options for each list item, and they are case sensitive.

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Edit Property Mapping** for the configured external tool. The **Edit Property Mapping** dialog box opens.
5. Select an external requirement type from the **Requirement types** list. All custom requirements of that type are then displayed below in the selection box.
6. Select the custom requirement property for which you are establishing mapping.
7. From the list box on the right, select the Silk Central custom property to establish mapping to the external custom property you have selected.
8. Click **Add Mapping** to map the requirements. The results are displayed in the **Custom property mapping** box.
9. The **System property mapping** box displays the two pre-configured mappings for requirement name and description, which cannot be removed.
10. Click **OK** to save your changes.

Deleting Property-Mapping Value Pairs

To delete a property-mapping value pair:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Edit Property Mapping** for the configured external tool. The **Edit Property Mapping** dialog box opens.

5. Select the property-mapping value pair in the **Custom property mapping** select box.
6. Click **Remove Mapping**.
7. Click **OK** to save your changes.

Disabling Requirements Management Integration

To disable requirements management integration configuration:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click the **Disable Configuration** button of the requirements management tool for which you want to disable integration.

All integration data and functionality is disabled, but not deleted from the project.

Removing Requirements Management Integration

To remove requirements management integration:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Remove Configuration** of the requirements management tool for which you want to remove integration.

This button is only enabled if the configuration is disabled.

The **Remove External Association** dialog box displays.

5. Click **Yes**.

All related data is removed from the database.

Issue Tracking Profile Integrations

Issue tracking profiles enable Silk Central to integrate with external issue tracking systems.

Additional issue tracking systems can be configured by installing a custom plug-in. For additional information, see *Issue Tracking Integration*. Defining issue tracking profiles allows you to link tests within the **Tests** area to issues in third-party issue-tracking systems. Linked issue states are updated periodically from the third-party issue tracking system.

Mapping Issue States

After defining a new issue tracking profile, you should map the existing issue states of the external issue tracking system to the issue states of Silk Central. Doing this enables Silk Central to list issues correctly when querying internal and external issues.

To map issue states:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Issue Tracking** tab. The **Issue Tracking** page opens, listing all of the issue tracking profiles that have been created for the system.
3. In the **Actions** column of the issue tracking profile you want to edit, click **Edit mapping of issue tracking profile** The **Edit Status Mapping** dialog box opens.
4. Map internal issue states to corresponding external issue states by selecting the respective entries from the list boxes.

The existing issue states of the external issue tracking software are listed in the **External** column. The internal issue states of Silk Central are available in the list boxes in the **Internal** column.

5. Once you have mapped each external issue state to an internal state, click **OK** to save your settings, or click **Cancel** to abort the operation.

Atlassian Jira Software

This section describes how to configure Atlassian Jira Software (Jira) issue tracking profiles to integrate with Silk Central.

The Jira plug-in relies on the Silk Central Java API for integration, using Jira's REST API.

Versions that are currently supported for integration with Silk Central:

- Atlassian Jira Software 6, 7, 8. Versions 6 and 7 are supported but not tested.
- Atlassian Jira Software Cloud. Not supported are *next-gen* projects and projects residing in a cloud instance containing *next-gen* projects.

If you assign an external Jira issue to a test, you can enter the issue ID either with or without the project key. For example: PROJECT-13 or just 13.

Silk Central supports the following Jira field types:

- **Text Field (< 255 characters) / Free Text Field (unlimited text)**
- **Select List (single choice) / Select List (multiple choices)**
- **Date Picker / Date Time Picker**



Note: These types are provided as text fields and need to be formatted like in Jira: **Date Picker:** dd/MMM/yy (for example 24/Mar/16), **Date Time Picker:** dd/MMM/yy h:mm a (for example 24/Mar/16 4:11 AM)

- **Group Picker (single group) / Group Picker (multiple groups)**
- **Project Picker (single project)**
- **User Picker (single user) / User Picker (multiple users)**
- **Version Picker (single version) / Version Picker (multiple versions)**

Other Jira field types are rendered as single-line text fields in Silk Central.

The following Jira system fields are not supported: Attachment

In the dialogs to create issues, required custom fields and required standard fields are marked with an asterisk (*).

Adding Atlassian Jira Software Issue Tracking Profiles

To add a Jira issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **Atlassian Jira Software** from the **Type** list.
6. Type a valid **Username** and **Password**. The user name needs to be a valid Jira user profile name. These credentials will be used to access your Jira issue tracking system.



Important: Use the credentials of a Jira account with sufficient permissions, as missing permissions in Jira can cause inconsistencies when the plug-in attempts to add or update issues

through this account. The Jira account requires the following permissions for each project that is used in Silk Central:

- Browse Projects
- Create Issues
- Assignable User
- Modify Reporter

Additionally, the Jira account needs to be able to access (log in to) your Jira application.

Use the following credentials to integrate with Jira cloud:

- **Username:** A Jira user profile name.
- **Password:** A Jira API token. For more information on how to generate the API token, refer to the Jira documentation.

Use the following credentials to integrate with an on-premise Jira instance:

- **Username:** A Jira user profile name.
 - **Password:** A Jira password.
7. In the **URL** field, type the host name of your Jira server and the port that is used to connect to the server.
 8. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
 9. Optional: Select **true** from the **Show all fields** list to display all available Jira fields in the issue dialog, otherwise only the required fields are displayed. If true is selected, the Jira user defined for the issue tracking profile has to have Jira administration permissions.
 10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing Atlassian Jira Software Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing Atlassian Jira Software issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Type a valid **Username** and **Password**. The user name needs to be a valid Jira user profile name. These credentials will be used to access your Jira issue tracking system.



Important: Use the credentials of a Jira account with sufficient permissions, as missing permissions in Jira can cause inconsistencies when the plug-in attempts to add or update issues through this account. The Jira account requires the following permissions for each project that is used in Silk Central:

- Browse Projects
- Create Issues
- Assignable User
- Modify Reporter

Additionally, the Jira account needs to be able to access (log in to) your Jira application.

Use the following credentials to integrate with Jira cloud:

- **Username:** A Jira user profile name.
- **Password:** A Jira API token. For more information on how to generate the API token, refer to the Jira documentation.

Use the following credentials to integrate with an on-premise Jira instance:

- **Username:** A Jira user profile name.
- **Password:** A Jira password.

7. In the **URL** field, edit the hostname of your Jira server and the port that is used to connect to the server.
8. To change the Jira project, click **Load Project** to load all projects from the server and update the **Project** list box, then select a project from the **Project** list box.
9. Optional: Select **true** from the **Show all fields** list to display all available Jira fields in the issue dialog, otherwise only the required fields are displayed. If true is selected, the Jira user defined for the issue tracking profile has to have Jira administration permissions.
10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Bugzilla

This section describes how to configure Bugzilla issue tracking profiles to integrate with Silk Central.

The Bugzilla plug-in relies on the Silk Central Java API for integration. Silk Central communicates with Bugzilla through the XML-RPC Bugzilla Web-service introduced with Bugzilla 3.0 by using the Redstone XML-RPC library. To enable Bugzilla integration, ensure that you have installed a SOAP::Lite package for the web server. For example, for the Debian distribution install the `libsoap-lite-perl` package.

Versions that are currently supported for integration with Silk Central: Bugzilla 4.4.13, 5.0.4

Adding Bugzilla Issue Tracking Profiles

To add a Bugzilla issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.

4. Type a **Description** for the new profile.
5. Select **Bugzilla** from the **Type** list box.
6. Type a valid **Username** and **Password**.

These credentials will be used to access the issue tracking system.

7. Enter the URL of your Bugzilla installation. For example, `http://bugzillaserver/bugzilla`.
8. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

9. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing Bugzilla Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing Bugzilla issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Edit the **Username** and **Password**.

These credentials are used to access your issue tracking system.
7. Edit the **URL** of your Bugzilla installation.
8. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

9. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

IBM Rational ClearQuest

This section describes how to configure IBM Rational ClearQuest issue tracking profiles to integrate with Silk Central.

IBM Rational ClearQuest products provide flexible defect/change tracking and automated workflow support. The two key products are IBM Rational ClearQuest (ClearQuest) and IBM Rational ClearQuest MultiSite (ClearQuest Multisite). To work with ClearQuest profiles, you must have the ClearQuest client software installed on the computer where the Silk Central front-end server is running. For detailed information about installing ClearQuest, refer to the ClearQuest documentation.

Versions that are currently supported for integration with Silk Central: IBM Rational ClearQuest 8.0

Adding ClearQuest Issue Tracking Profiles

To add a ClearQuest issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **IBM Rational ClearQuest** from the **Type** list box.
6. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
7. Enter the **Repository Info** of your ClearQuest installation.
This is the database name that is defined in the ClearQuest client software. The name of the sample database is `SAMPL`.



Note: It is not possible to use more than one ClearQuest schema repository. Silk Central will connect to the default ClearQuest repository.

8. Specify the **Record Type**, which is the issue type of ClearQuest.
When entering an issue in Silk Central, ClearQuest will save the issue with the issue type you define in this field. The default issue type is `Defect`.
9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

10. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing ClearQuest Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing ClearQuest issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Edit the **Username** and **Password**.
These credentials are used to access your issue tracking system.
7. Edit the **Repository Info** of your ClearQuest installation.

This is the database name that is defined in the ClearQuest client software. The name of the sample database is `SAMPL`.



Note: It is not possible to use more than one ClearQuest schema repository. Silk Central will connect to the default ClearQuest repository.

8. Change the **Record Type**, which is the issue type of ClearQuest.

When entering an issue in Silk Central, ClearQuest saves the issue with the issue type you define in this field.

9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

10. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.

- Click **Yes** to proceed with the related *Mapping Issue States* procedure.
- Click **No** to map issue states later.

Issue Manager

Issue Manager, the issue-tracking tool of Silk Central, is fully integrated with Silk Central, enabling you to correlate issues with system requirements and executed tests.

Test issues can be added and managed in the menu through **Tests > Details View > Issues**. For more information, see the Issue Manager topics in this Help.

Adding Issue Manager Issue Tracking Profiles

To add an Issue Manager issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **Issue Manager** from the **Type** list box.
6. Check the **Local Repository** check box to use the integrated Issue Manager profile that is part of your Silk Central instance. To connect to an external Issue Manager system, uncheck this option, in which case you will need to provide valid login credentials to access the system.
7. Type the **Issue Manager URL** of your Issue Manager installation. This is the URL you use to login to Issue Manager, though without the `login` extension at the end. Leave the default URL if you selected the **Local Repository** option.

For example, if your standard Issue Manager URL is `http://IssueManager/login`, then the correct service URL is `http://IssueManager`.

8. *Optional:* If Issue Manager should communicate with the front-end server through a different URL than the one that users use to access the Web user interface, type this URL into the **Web Service URL** field. This is required for example when users access the Web user interface through a proxy, while Issue Manager needs the internal URL or IP address of the front-end server.
9. Proceed as follows:

1. Click **Load Projects**. This action will populate the **Project** list box with all initialized Issue Manager projects to which the specified user has access to. Note that only those projects display for which Issue Manager user groups have been defined, and the defined user is a member of at least one user group.

2. Select the **Project** where Issue Manager issues are maintained.



Caution: We recommend not to use identical projects for Issue Manager and Silk Central, as this limits flexibility in working with both tools on different future projects.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing Issue Manager Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing Issue Manager issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Check the **Local Repository** check box to use the integrated Issue Manager profile that is part of your Silk Central instance. To connect to an external Issue Manager system, uncheck this option, in which case you will need to provide valid login credentials to access the system.
7. Edit the **Issue Manager URL** of your Issue Manager installation if the location has changed.

This is the URL you use to login to Issue Manager, though without the `login` extension at the end. For example, if your standard Issue Manager URL is `http://IssueManager/login`, then the correct service URL would be `http://IssueManager`.
8. *Optional:* If Issue Manager should communicate with the front-end server through a different URL than the one that users use to access the Web user interface, type this URL into the **Web Service URL** field. This is required for example when users access the Web user interface through a proxy, while Issue Manager needs the internal URL or IP address of the front-end server.
9. To change the Issue Manager project, click **Load Projects** to load all projects from the server and update the **Project** list box, then select a project from the **Project** list box.
10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

SAP Solution Manager Integration

To purchase the SAP Solution Manager Plug-In, please contact your sales representative.

SAP Solution Manager facilitates technical support for distributed systems, covering solution deployment, operation, and continuous improvement. SAP Solution Manager is a centralized, robust application management and administration solution, that combines tools, content, and direct access to SAP to increase the reliability of solutions and lower total cost of ownership. For additional information on the SAP Solution Manager, refer to the SAP Solution Manager documentation.

The integration between Silk Central and SAP Solution Manager allows for the synchronization of SAP Solution Manager business processes and incidents as Silk Central requirements and issues.

System Requirements for SAP Solution Manager Integration

The following are the system requirements for the SAP Solution Manager integration plugin:

- SAP Solution Manager 7.2.
- Silk Central 19.0 or later.
- The *Silk Central Integration Hub for SAP Solution Manager* must be installed in SAP Solution Manager.

Configuring SAP Solution Manager

Configure SAP Solution Manager to enable the integration with Silk Central.

For additional information, see

[Configuration_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#) and [User_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#), which are included in the SAP Solution Manager plug-in.

For this integration to get valid links from SAP Solution Manager to Silk Central, the front-end server machine of Silk Central must have a valid fully qualified name.

Adding SAP Solution Manager Issue Tracking Profiles

To add a SAP Solution Manager issue tracking profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Issue Tracking** tab. The **Issue Tracking** page opens, listing all of the issue tracking profiles that have been created for the system.
3. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Type a **Description** for the new profile.
6. Select **SAP Solution Manager** from the **Type** list box.
7. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
8. Type the **URL** of the Z-SCTM_ADAPTER web service of SAP Solution Manager.
9. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
10. Click **Load Default Incident Type** to load all possible incident types and populate the **Default Incident Type** list box.
11. Select an incident type from the **Default Incident Type** list box. If you are creating links to existing incidents in SAP Solution Manager and you do not specify the incident type, this type will be used for it.
12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

13. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Enabling Requirements Integration with SAP Solution Manager

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **SAP Solution Manager** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.
8. Enter the **URL** of the `Z_SCTM_ADAPTER` web service of SAP Solution Manager.
9. Enter the **Username** and **Password**.
10. Click **Test Connection** to confirm that the host and user credentials are correct. If the settings are correct, a **Test connection was successful** message displays.
11. Click **Load** to load all available solution and branch combinations and select the correct one.

Silk Central stores such combinations of solutions and branches as projects.
12. *Optional:* Check the **Enable creation of unassigned requirements** check box to enable creation and editing of unmapped requirements in Silk Central projects that are configured for integration.
13. Click **OK** to confirm and close the dialog.

Microsoft Team Foundation Server

This section describes how to configure Team Foundation Server (TFS) issue tracking profiles to integrate with Silk Central.

Versions that are currently supported for integration with Silk Central: Team Foundation Server 2015, 2017

Installing a Team Foundation Server Web Service Proxy

To communicate with a TFS, you need to install a Team Foundation Server Web Service Proxy as an interface to the TFS.

To install a TFS proxy service:

1. Install IIS and the Web Deployment Tool, for example with the Web Platform Installer from <http://www.microsoft.com/web/downloads/platform.aspx>.
2. Open the Windows Features dialog and ensure that ASP .NET 4.x is checked.

In Microsoft Windows 10, the feature is located under:

Internet Information Services/World Wide Web Services/Application Development Features

3. Download the Team Foundation Server Web Service Proxy from **Help > Tools**.
4. Unzip the downloaded package.
5. Open a command shell and type `DotNetTfsWebServiceProxy.deploy.cmd /Y` to install the proxy service. If an error message box displays, stating that ASP.NET 4 is required, refer to <http://msdn.microsoft.com/en-us/library/k6h9cz8h.aspx>.
6. Open the IIS Manager.
7. Verify that the new website exists.
8. In the root folder of the virtual directory, open the `Web.config` file.
9. In the `appSettings` section of the file, modify the value of the key `WorkItemTrackingCacheRoot` to a local directory.

For example:

```
<appSettings>
  <add key="WorkItemTrackingCacheRoot" value="C:\temp" />
</appSettings>
```

10. If the directory you have specified does not exist, create it.
11. Grant **Everyone** full rights on the specified directory.

For additional information on deploying an ASP.NET Web Application, refer to <http://msdn.microsoft.com/en-us/library/ee942158.aspx>.



Note: If you are upgrading Silk Central from version 16.5 or earlier, you need to re-deploy the TFS proxy service. Delete the existing TFS proxy service, then continue with step 3 above. When you are done, restart IIS.

Adding TFS Issue Tracking Profiles

You need to configure a Team Foundation Server Web Service Proxy, which is provided by Micro Focus to enable Silk Central to access TFS issues.



Important: This is not the Team Foundation Server Proxy provided by Microsoft.

To add a TFS issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **Team Foundation Server** from the **Type** list box.
6. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
7. Type the **Domain** of the TFS user.
8. Enter the URL of your TFS installation.
For example, `http://tfsserver:8080/tfs`.
9. Type the **Collection** to which your project belongs.
For example `DefaultCollection`.
All projects in the collection are listed in the **Projects** list box.
10. Select the **Project** from the list box.
11. Type the URL of your TFS proxy. For example `http://tfsproxyserver/DotNetTfsWebServiceProxy_deploy/TfsWebServiceProxy.asmx`.

12. Click **Load Work Item Type**. The **Work Item Type** list box is populated with the available work item types.
13. Select the **Work Item Type** from the list box.
14. Click **Load Initial State**. The **Initial State** list box is populated with the states that are allowed for the selected work item type.
15. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

16. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing TFS Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing TFS issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Edit the **Username** and **Password**.

These credentials are used to access your issue tracking system.
7. Edit the **Domain** of the TFS user.
8. Edit the **URL** and **Collection** of your TFS installation.
9. Select a different **Project**.
10. Edit the URL and port of your TFS proxy.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

StarTeam Issue Tracking Profiles

This section describes how to configure StarTeam issue tracking profiles to integrate with Silk Central.

StarTeam is a software change management and configuration management tool that enables coordination and management of the software delivery process.

To work with StarTeam profiles and use the go-to-link functionality for change requests in StarTeam, you must have the StarTeam Cross-Platform Client software installed on the computer where the browser is running.

Versions that are currently supported for integration with Silk Central: StarTeam 15.x, 16.x, 17.0

Adding StarTeam Issue Tracking Profiles

To add a StarTeam issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **StarTeam** from the **Type** list box
6. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
7. Type the **Hostname** of your StarTeam server and the **Port** that is used to connect to the server. If this setting has not been changed, use the default port 49201.
8. Specify the type of **Encryption** that the profile supports.
9. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
10. Click **Load View** to load all views for the selected project and populate the **View** list box, then select a view from the **View** list box.
11. Select the issue type from the **Item Type** list.



Note: Selecting an item type other than **ChangeRequest** only works if this type has a StarTeam workflow configured.

12. Click **Load Status Field** to load all enumeration fields for change requests and populate the **Status Field** list box, then select a status field from the **Status Field** list box.
If you are using a custom workflow in StarTeam, this field is the workflow driver field in StarTeam that maps to the Silk Central issue state.
13. Select a type of link from the **Link Type** list box.

starteam:// External ID links on the **Issues** tab will open the cross platform client.

http:// External ID links on the **Issues** tab will open the issue in the StarTeam web UI.

14. If you selected `http://` in the **Link Type** list box, enter the web server address of the StarTeam web UI in the **WebServer** field.
15. Select **Yes** or **No** in the **Workflow** field. Select **Yes** to show the required fields that are specified in the workflow of the selected **View**.
16. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

17. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing StarTeam Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing StarTeam issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Edit the **Username** and **Password**.
These credentials are used to access your issue tracking system.
7. Edit the **Hostname** of your StarTeam server and the **Port** that is used to connect to the server.
8. Modify the type of **Encryption** that the profile supports.
9. To change the StarTeam project, click **Load Project** to load all projects from the server and update the **Project** list box, then select a project from the **Project** list box.
10. To change the view, click **Load View** to load all views for the selected project and populate the **View** list box, then select a view from the **View** list box.
11. Select the issue type from the **Item Type** list.



Note: Selecting an item type other than **ChangeRequest** only works if this type has a StarTeam workflow configured.

12. To change the workflow driver field, click **Load Status Field** to load all enumeration fields for change requests and populate the **Status Field** list box, then select a status field from the **Status Field** list box.
If you are using a custom workflow in StarTeam, this field is the workflow driver field in StarTeam that maps to the Silk Central issue state.
13. Select a type of link from the **Link Type** list box.

 - starteam://** External ID links on the **Issues** tab will open the cross platform client.
 - http://** External ID links on the **Issues** tab will open the issue in the StarTeam web UI.
14. If you selected `http://` in the **Link Type** list box, enter the web server address of the StarTeam web UI in the **WebServer** field.
15. Select **Yes** or **No** in the **Workflow** field. Select **Yes** to show the required fields that are specified in the workflow of the selected **View**.
16. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

17. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Deleting Issue Tracking Profiles

To delete an issue tracking profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Issue Tracking** tab. The **Issue Tracking** page opens, listing all of the issue tracking profiles that have been created for the system.
3. In the **Actions** column of the issue tracking profile you wish to delete, click **Delete issue tracking profile** A confirmation dialog box displays.

4. Click **Yes**.

Issue Tracking Profiles Page

Project:<Project Name> > Project Settings > Issue Tracking

Use this page to configure profiles for the integration of external issue tracking systems into Silk Central. Click **New Profile** to create a new issue tracking profile. For each issue tracking profile, the page displays the following columns:

| Column | Description |
|-----------------|---|
| Name | The name of the profile as it displays in the Silk Central GUI and in reports. Click the name to edit a profile. |
| Type | The external issue tracking system. |
| Login | The login name with which Silk Central connects to the issue tracking system. |
| Repository Info | The physical location of the issue tracking system. Hostname or URL. |
| Created On | Date when the issue tracking profile was created. |
| Created By | The user who created the issue tracking profile. |
| Actions | The following actions can be performed on an issue tracking profile: <ul style="list-style-type: none">• Delete issue tracking profile• Edit mapping of issue tracking profile |

Source Control Profiles

Source control profiles enable Silk Central to integrate with external source control systems. Defining source control profiles allows you to define where Silk Central's execution servers should retrieve program sources for test execution.

You need double the amount of free disk space on the execution server to accommodate the source files because Silk Central checks out the source control tree and then generates a working directory with the executable source files.

Additional source control systems can be configured by installing a custom plug-in. For detailed information, refer to the *Silk Central API Help*. The following source control systems are currently supported by Silk Central "out of the box":

AccuRev

This section describes how to configure AccuRev source control profiles.

AccuRev is a software configuration management tool which addresses complex parallel and distributed development environments with its stream-based architecture, to accelerate your development process and improve asset reuse.

AccuRev versions that are currently supported for integration with Silk Central: AccuRev 7.2

Adding AccuRev Source Control Profiles

To create a AccuRev source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.

3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **AccuRev** from the **Source control system** list box.
6. Type a valid AccuRev **Username** and **Password**, if required.
7. Type the **AccuRev server** name of the AccuRev server you want to access. If you are not using AccuRev's default port, type `<server name>:<port number>`.
8. Type the name of the AccuRev **Stream**, for example `backing stream`.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.
The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing AccuRev Source Control Profiles

To modify a AccuRev source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the **Username** and **Password**.
These credentials are used to access your repository.
6. Edit the **AccuRev server** name of the AccuRev server you want to access. If you are not using AccuRev's default port, type `<server name>:<port number>`.
7. Edit the name of the AccuRev **Stream**, for example `backing stream`.
8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Edit the **Project path** you want this profile to use.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Apache Commons Virtual File System

This section describes how to configure Virtual File System (VFS) source control profiles.

A VFS is an abstraction layer on top of a more concrete file system. The purpose of a VFS is to allow client applications to access different types of concrete file systems in a uniform way. Apache Commons VFS provides a single API for accessing various file systems. It presents a uniform view of the files from various sources. The protocols that are currently supported for VFS by Silk Central are:

Protocol Description

| | |
|-------------|---|
| http | Copies the given file. This protocol type is also supported for copying and unpacking ZIP, JAR, or other zipped files. It is required to specify a .zip file on a http server. For example, <code>zip:http://myTestServer/myTests.zip</code> . The .zip file will be extracted on the execution server. |
| ftp | Copies the given file. This protocol type is also supported for copying and unpacking ZIP, JAR, or other zipped files. |
| smb | Server Message Block (smb) copies all files and folders. This protocol can be used instead of a UNC profile. For example, the VFS smb path <code>smb://server-name/shared-resource-path</code> is equivalent to the UNC path <code>\\server-name\shared-resource-path</code> . |



Note: When you create a new ProcessExecutor test that uses VFS for source control, you need to specify the complete path to the executable in the **Executable Name** field.

Adding VFS Source Control Profiles

To create a VFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **VFS** from the **Source control system** list box.
6. Type the URL of the VFS **Repository** you want to access. This field is case sensitive.

Specify the appropriate protocol type in the URL:

| | |
|-------------|---|
| FTP | <code>ftp://<FTP server URL></code> |
| HTTP | <code>http://<HTTP server URL></code> |
| SMB | <code>smb://<Samba server URL></code> |



Note: HTTP, FTP and SMB are also supported for zipped files. In order to point to a zipped file the URL must be adjusted to `<zipped file type>:<protocol>://<server URL pointing to zipped file>` to include the type of the zipped file. For example, `zip:http://193.80.200.135/<path>/archive.zip` or `jar:http://193.80.200.135/<path>/archive.jar`.

7. Type a valid VFS **Username** and **Password**.

These credentials will be used to access your VFS repository. The SMB protocol allows including the domain name in the username in the following form: `domain/username`.

8. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing VFS Source Control Profiles

To modify a VFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.

4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Edit the URL of the VFS **Repository** you want to access.

 **Note:** This field is case sensitive.

6. Edit the **Username** and **Password**.

These credentials are used to access your repository.

7. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

8. Edit the **Project path** you want this profile to use.

9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Git

This section describes how to configure Git source control profiles.

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed. Every Git working directory is a full-fledged repository with complete history and full version tracking capabilities, not dependent on network access or a central server.

Adding Git Source Control Profiles

To create a Git source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **Git** from the **Source control system** list box.
6. Type the URL of the Git **Repository** you want to access.

 **Note:** This field is case sensitive.

7. Type a valid Git **Username** and **Password**, if required.
8. *Optional:* Specify a **Branch** to check out a specific branch. Leave this field empty to check out the master branch.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, C:\TempSources\.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

 **Tip:** If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan. See *Creating Execution Plans* for detailed information. This will overwrite the specified **Branch** in the Git source control profile.

Editing Git Source Control Profiles

To modify a Git source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.

3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.

4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Type the URL of the Git **Repository** you want to access.



Note: This field is case sensitive.

6. Edit the **Username** and **Password**.

These credentials are used to access your repository.

7. *Optional:* Specify a **Branch** to check out a specific branch. Leave this field empty to check out the master branch.

8. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

9. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.



Tip: If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan. See *Creating Execution Plans* for detailed information. This will overwrite the specified **Branch** in the Git source control profile.

Microsoft Team Foundation Server

This section describes how to configure Team Foundation Server (TFS) source control profiles.

TFS versions that are currently supported for integration with Silk Central: Team Foundation Server 2015, 2017



Note: To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).

Adding TFS Source Control Profiles

To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).

 **Note:** If your TFS project has a Git repository configured, you have to create a Git source control profile instead of a TFS source control profile. Use . as the user name and a token that you have generated in TFS as the password for the Git profile. For additional information on creating a Git source control profile, see [Adding Git Source Control Profiles](#).

To create a TFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **Team Foundation Server** from the **Source control system** list box.
6. Enter the **URL** of your TFS installation.
For example, `http://tfsserver:8080/tfs`.
7. Type the **Domain** of the TFS user.
8. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.
The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing TFS Source Control Profiles

To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).

To modify a TFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.

5. Edit the URL of the TFS **Repository** you want to access.
6. Edit the **Username** and **Password**.

These credentials are used to access your repository.

7. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

8. Edit the **Project path** you want this profile to use.
9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Silk Test Workbench

This section describes how to configure Silk Test Workbench source control profiles.

Silk Test Workbench is an automated testing tool that accelerates the functional testing of complex applications. Silk Test Workbench provides support for testing applications developed in a wide variety of development tools including Java, .NET, browser-based web applications, and COM components, including both ActiveX controls and automation objects. With Silk Test Workbench, you can record user sessions with your applications to create tests, enhance the test by adding validation and test logic, and play back tests to ensure that the applications work as expected.



Note: To add a Silk Test Workbench source control profile, the Silk Test Workbench client needs to be installed on every execution server on which you want to use the source control profile.

Adding Silk Test Workbench Source Control Profiles

To create a Silk Test Workbench source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.

4. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.

5. Select **Silk Test Workbench Test** from the **Source control system** list box.
6. Select a database type from the **Database type** list box, either SQL Server or Oracle.



Note: The Microsoft Office Access database type is not supported. When connecting to a Silk Test Workbench Oracle database, there is a Silk Test Workbench requirement that the TNS name must be the same as the host server name.

7. Enter the name of the Silk Test Workbench database server in the **Database server** field.
8. Enter the name of the Silk Test Workbench database in the **Database name** field.
9. Enter the port number of the Silk Test Workbench database server in the **Database port** field.
10. Enter the Silk Test Workbench database schema or owner name in the **Database schema/owner name** field.

11. Enter the name of the database user in the **Database user** field.
12. Enter the password for the database user in the **Database password** field.
13. *Optional:* Enter the name and password of a valid Silk Test Workbench user in the **Silk Test Workbench User name** and **Silk Test Workbench password** fields. If these fields are left empty, the Windows user that runs the execution server service is used.
14. Click **Retrieve**. All projects are listed in the **Projects** list.
15. Select one or more projects.
16. Click **OK**.

Editing Silk Test Workbench Source Control Profiles

To modify a Silk Test Workbench source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the Silk Test Workbench **Database type**.
 **Note:** The Microsoft Office Access database type is not supported.
6. Edit the other settings as required.
7. Click **OK**.

StarTeam

This section describes how to configure StarTeam source control profiles.

StarTeam promotes team communication and collaboration through centralized control of all project assets. Protected yet flexible access ensures that team members can work whenever and wherever they like through an extensive choice of Web, desktop, IDE, and command-line clients. StarTeam offers a uniquely comprehensive solution that includes integrated requirements management, change management, defect tracking, file versioning, threaded discussions, and project and task management.

StarTeam versions that are currently supported for integration with Silk Central: StarTeam 15.x, 16.x, 17.0

Adding StarTeam Source Control Profiles

To create a StarTeam source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **StarTeam** from the **Source control system** list box.
6. Type the **Hostname** of your StarTeam server.
7. Type the **Port** that is to be used to connect to the StarTeam server.
If the port is not changed, use the default port 49201.

8. Type a valid **Username** and **Password**.

These credentials will be used to access your repository.

9. Specify if the profile supports **Encryption**.

10. *Optional*: Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

11. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing StarTeam Source Control Profiles

To modify a StarTeam source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .

2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.

3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.

4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Choose from the following options:

- Edit the **Hostname** of your StarTeam server.
- Edit the port that is to be used to connect to the StarTeam server. If the port is not changed, use the default port 49201.
- Specify if the profile supports **Encryption**.

6. Edit the **Project path** you want this profile to use.

7. Edit the **Username** and **Password**.

These credentials are used to access your repository.

8. *Optional*: Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.

9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Subversion

This section describes how to configure Subversion (SVN) source control profiles.

If you set a Source Control Label on the execution plan, you can check out specific Subversion branches or tags. Branches and tags are encoded in the Subversion URL. For example, `http://MyHost/svn/MyApp/trunk` is a Subversion URL and `http://MyHost/svn/MyApp/tags/build1012` is a Subversion tag.

To check out branches and tags in Silk Central, the main repository URL that is specified in the source control profile must contain the `trunk` directory. The `trunk` directory in the Subversion URL is then replaced with the label that is specified for the execution plan.

Adding Subversion Source Control Profiles

To create a Subversion source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.

5. Choose **Subversion** from the **Source control system** list box.
6. Type the URL of the Subversion **Repository** you want to access.

If you do not know the URL of the repository, consult your Subversion administrator.



Note: To use SSH, add `ssh` to the URL, for example `svn+ssh://<hostname>:<port>`.

7. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
8. Type a valid **SSH username** and **SSH password** or **SSH keyfile**.
These credentials are used to access the SSH server. The password overrides the keyfile, so if you only have a keyfile, leave the **SSH password** field empty. If you use a keyfile, the path to the keyfile must be valid on every execution server that uses the source control profile.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing Subversion Source Control Profiles

To modify a Subversion source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.

4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Edit the **URL** of the Subversion repository you want to access.

If you do not know the URL of the repository, please consult your Subversion administrator.



Note: If you cannot check out files, when editing the URL of the Subversion **Repository**, delete the source control mirrors directory on your execution server. For example, `C:\ProgramData\SilkCentral\SrcCtrlMirrors`.



Note: To use SSH, add `ssh` to the URL, for example `svn+ssh://<hostname>:<port>`.

6. Edit the **Username** and **Password**.

These credentials are used to access your repository.

7. Type a valid **SSH username** and **SSH password** or **SSH keyfile**.

These credentials are used to access the SSH server. The password overrides the keyfile, so if you only have a keyfile, leave the **SSH password** field empty. If you use a keyfile, the path to the keyfile must be valid on every execution server that uses the source control profile.

8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Edit the **Project path** you want this profile to use.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Silk TestPartner

This section describes how to configure TestPartner source control profiles.

TestPartner is a Micro Focus product that tests Web- and Microsoft Windows-based applications that use Microsoft technologies. TestPartner records user actions to quickly produce powerful tests. Each recorded test displays as a series of actions in clear, concise steps that can be easily understood by all testers, from novice to expert. You can record user sessions with the application, add validation functions, and replay the sessions later to ensure that the application works as expected.



Note: To add a TestPartner source control profile, the TestPartner client needs to be installed on every execution server on which you want to use the source control profile.

Adding TestPartner Source Control Profiles

To create a TestPartner source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **TestPartner** from the **Source control system** list box.



Note: When connecting to a TestPartner Oracle database, there is a TestPartner requirement that the TNS name must be the same as the host server name.

6. Select a database type from the **Database type** list box, either SQL Server or Oracle.
7. Type the name of the database server in the **Database server** field.
8. Type the name of the database in the **Database name** field.
9. Type the port number of the database server in the **Database port** field.
10. Type the database schema or owner name in the **Database schema/owner name** field.
11. Type the name of the database user in the **Database user** field.
12. Type the password for the database user in the **Database password** field.
13. Type a valid **Username** and **Password**.

These credentials will be used to access your repository.

14. Use the **Project path** field to filter which scripts from the database are available to use as tests. Click **Browse** to display the **Select Project Path** dialog box.

The **Select Project Path** dialog box provides a tree of three levels that you can choose from:

- First level** The entire TestPartner database. Note that if you select this level, the **Project path** field will remain blank.
- Second level** The TestPartner project.
- Third level** The script type within a specified TestPartner project.

Select one of the preceding options and click **OK**. The path will be added to the **Project Path** field.

15. Click **OK**.

Editing TestPartner Source Control Profiles

To modify a TestPartner source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Choose from the following options:

- Edit the TestPartner **Database type**.
 - Edit the TestPartner **Database server**.
 - Edit the TestPartner **Database name**.
 - Edit the TestPartner number of the TestPartner database server.
 - Edit the TestPartner **Database schema/owner name**.
 - Edit the TestPartner **Database user**.
 - Edit the TestPartner **Database password**.
6. Edit the **Username** and **Password**.
These credentials are used to access your repository.
 7. Edit the **Project path** you want this profile to use.
 8. Click **OK**.

Universal Naming Convention

This section describes how to configure Universal Naming Convention (UNC) source control profiles.

Short for Universal Naming Convention or Uniform Naming Convention, UNC is a PC format for specifying the location of resources on a local area network (LAN). UNC uses the following format: `\\server-name\shared-resource-pathname`.

For example, to access the file `test.txt` in the directory `examples` on the shared server `silo`, you would write: `\\silo\examples\test.txt`.

You can also use UNC to identify shared peripheral devices, such as printers. The idea behind UNC is to provide a format so that each shared resource can be identified with a unique address.

UNC is only supported on Microsoft Windows operating systems. If you plan to use a non-Windows execution server you can use the Apache Commons VFS source control profile instead.

Adding UNC Source Control Profiles

To create a UNC source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
 2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
 3. Click **New Profile** to open the **New Source Control Profile** dialog box.
 4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
 5. Select **UNC** from the **Source control system** list box.
 6. Type the **UNC path** that you want to access.
This is the path to the location where your test sources are located.
-  **Tip:** To avoid checking out unnecessary files, specify the folder at a hierarchy level where the sources reside that are actually required for test execution.
7. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`. If you do require a static working folder, make sure that it is not located within the specified **UNC path**, as this will lead to recursive checkouts.

8. Type a valid **UNC username** and **UNC password**.
These credentials are required to access the UNC path of the configuration file.

9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing UNC Source Control Profiles

To modify a UNC source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.

4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Type the **UNC path** that you want to access.

This is the path to the location where your test sources are located.



Tip: To avoid checking out unnecessary files, specify the folder at a hierarchy level where the sources reside that are actually required for test execution.

6. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`. If you do require a static working folder, make sure that it is not located within the specified **UNC path**, as this will lead to recursive checkouts.

7. Edit the **UNC username** and **UNC password**.

These credentials are required to access your configuration file's UNC path.

8. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Deleting Source Control Profiles

To remove a source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. In the **Actions** column of the issue tracking profile you wish to delete, click **Delete source control profile** A confirmation dialog box displays.
4. Click **Yes**. You are returned to the **Source Control** page.

Source Control Profiles Page

Project:<Project Name> > **Project Settings** > **Source Control**

Use this page to configure profiles to integrate external source control systems with Silk Central. Click **New Profile** to create a new source control profile. For each source control profile, the page displays the following columns:

| Item | Description |
|----------------|--|
| Name | The name of the profile as it displays in the Silk Central GUI and in reports. Click the name to edit a profile. |
| Type | The external source control system. |
| Working Folder | Local or mapped working folder to which temporary sources are checked out to. |
| Created On | Date when the source control profile was created. |
| Created By | The user who created the source control profile. |
| Changed On | Date when the source control profile was last modified. |
| Changed By | The user who last modified the source control profile. |
| Actions | Delete source control profile |

Data Sources for Data-Driven Tests

Data-driven tests are tests that are derived from values in an existing data source, such as a spreadsheet or a database. The data sources are managed in a project-specific scope.



Tip: To acknowledge changes in your data source to Silk Central, you must synchronize your data source profile with your data source whenever your data source is updated or changed.

Data Source Limits

Data sources have specific limits to prevent Silk Central from getting into an unstable state due to excessive data generation.

Atomic vs. Non-Atomic Tests

For detailed information on the data-driven test types that are available in Silk Central, see [Atomic and Non-Atomic Data-Driven Tests](#).

Limits

Data sources for data-driven tests have the following default limits:

- 100 rows for non-atomic tests
- 1000 rows for atomic tests
- 20000 rows+cells for **XLSX** data sources (20000 is the amount of rows + the amount of cells. For example: If you have 3000 rows with 6 columns, the calculated amount is $3000 + (3000 * 6) = 21000$, so the limit is exceeded.)
- 220 characters for data set names (Excel sheet names/table names)
- 220 characters for column key names
- The concatenated length of all primary keys may not exceed 255 characters in length for JDBC, Excel, and CSV data sources
- 2000 characters per cell (longer strings will be truncated). Note that data-driven Silk Test Classic tests only accept data source strings in quotation marks. If the string length in a cell exceeds 2000 characters, the string is truncated and the trailing quotation mark is missing, which leads to an error.

Modifying the Default Limits

The default limits can be modified, if required. Modify these values with care though to avoid excessive data generation. Before attempting to modify these values, stop the application server and restart it again after you have saved your changes.

The row limits for atomic and non-atomic tests can be configured in the `TmAppServerHomeConf.xml` file. The default path for this file is `C:\Program Files (x86)\Silk\Silk Central 20.6\instance_<instance number>_<instance name>\Conf\AppServer` on the application server. Modify the values of the `Config/DataDriven/MaxRowsAtomic` and `Config/DataDriven/MaxRows` XML tags, respectively.

The XLSX cell limit can be configured in the `configuration.properties` file in the Excel data source plugin (`DS-Excel.zip`). The default path for this file is `C:\Program Files (x86)\Silk\Silk Central 20.6\instance_<instance number>_<instance name>\Plugins` on the application server. Modify the value of the `excel.importcelllimit` property accordingly.

Configuring JDBC Data Sources

 **Caution:** The concatenated length of all primary keys may not exceed 255 characters in length for JDBC data sources.

To configure a JDBC data source:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click **New Data Source** to open the **New Data Source** dialog box.
4. Type a **Name** for the data source.
5. From the **Data source type** list box, select **JDBC**.



Note: If you are setting up an ODBC data source, you need to manually insert your ODBC **Driver class** and **URL**. For example, **Driver class:** `sun.jdbc.odbc.JdbcOdbcDriver`, **URL:** `jdbc:odbc:MyDatabaseName`. You must also set up an ODBC data source in MS Windows in the **Administrative Tools**. For more information, refer to Microsoft Windows Help. If you have your front-end server and your application server on different machines, make sure that the name of your system data source set-up in Microsoft Windows is the same as the ODBC data source. These names are case-sensitive.

6. If you select **JDBC** as the **Data source type**, the **Driver class** field is populated automatically. In the **URL** field, replace the host name value, `<hostname>`, with the name of the computer that is hosting the data source and replace the database name value, `<databasename>`, with the name of the target database.
7. In the **Username** and **Password** fields, enter valid database credentials.
8. *Optional:* If you are working with a database that includes multiple tables, and you want to narrow down the data source to specific tables, you can browse to and select specific tables for inclusion:
 1. Click [...] next to the **Table filter** field.
 2. The **Select Table Filter** dialog box displays. Select the tables that you want included as your data source.
 3. Click **OK**.
9. *Optional:* **Key column** selection is used by tests to define which worksheet columns within a data source are used as primary key. This is helpful if your data source will undergo edits, for example when you add or remove rows within a worksheet. Even if your data source is edited, tests will still be able to identify which columns or rows should be used. Tests created from data-driven data sources use key column values in their names, rather than column numbers.

To configure a key column:

1. Click [...] next to the **Key column** field.
 2. The **Select Key Column** dialog box displays. Select a column from the column list that is to act as a key column.
 3. Click **OK**.
10. Click **OK** on the **New Data Source** dialog box.

Configuring Microsoft Excel or CSV Data Sources

 **Caution:** Excel worksheets using password protection can not be configured as data sources for Silk Central. Turn off a worksheet's password protection to use it as data source for data-driven testing. The concatenated length of all primary keys may not exceed 255 characters in length for Excel and CSV.

To configure a Microsoft Excel or CSV data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click **New Data Source** to open the **New Data Source** dialog box.
4. Type a **Name** for the data source.
5. From the **Data source type** list box, select **MS Excel** to configure a Microsoft Excel data source, or select **CSV** to configure a CSV data source.
6. From the **Source control profile** list box, select the pre-configured source control profile that hosts your data file. For detailed information regarding the configuration of source control profiles, see *Source Control Profiles*.
7. Click **Browse** to open the **Select Source Control Path** dialog box. Browse to and select a data source file of the selected type in your source control path.
8. *Optional:* MS Excel only. If you are working with an Excel spreadsheet that includes multiple worksheets, and you want to narrow down the data source to specific worksheets, you can browse to and select specific worksheets for inclusion. To do this:
 1. Click [...] next to the **Worksheet filter** field.
 2. The **Select Worksheet Filter** dialog box displays. Select the worksheets that you want included as your data source.
 3. Click **OK**.
9. *Optional:* **Key column** selection is used by tests to define which worksheet columns within a data source are used as primary key. This is helpful if your data source will undergo edits, for example when you add or remove rows within a worksheet. Even if your data source is edited, tests will still be able to identify which columns or rows should be used. Tests created from data-driven data sources use key column values in their names, rather than column numbers.

 **Note:** MS Excel only: If the data source includes multiple worksheets, only columns with identical names are available to be defined as key columns.

To configure a key column:

1. Click [...] next to the **Key column** field.
 2. The **Select Key Column** dialog box displays. Select a column from the column list that is to act as a key column.
 3. Click **OK**.
10. Click **OK** on the **New Data Source** dialog box.

Downloading Excel Files from a Data Source

 **Note:** Files can not be downloaded from JDBC and ODBC data sources.

To download an Excel file from a data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Download** icon in the **Actions** column of the data source you want to download from. The **File Download** dialog box displays.
4. Click **Open** to open the file immediately, or click **Save** to specify where on your local system you want to save the file to.

Synchronizing Data Sources

You must synchronize a data source each time it is changed or updated, if you want to make Silk Central aware of the changes. Synchronizing a data source propagates recent changes to associated tests.

To synchronize an updated data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Synchronize** icon in the **Actions** column that corresponds to your data source to propagate the updated file to the associated tests.
4. A confirmation dialog box displays, asking you to confirm the synchronization. Click **Yes** to synchronize all tests with the updated data source, or click **No** to abort the synchronization.



Caution: When you synchronize a data source, all running executions depending on this data source are aborted. Results of incomplete tests within these executions are lost.

5. Click **OK** on the success message dialog box.

Uploading Updated Excel Files to a Data Source



Note: Files can not be uploaded to JDBC and ODBC data sources.

To upload an updated Excel file to a data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Upload** icon in the **Actions** column of the data source you want to download from.
4. Click **Browse...** on the **Upload File** dialog box.
5. Select the updated Excel file that you want to replace the currently uploaded Excel file with. Click **Open**.
6. Click **OK** on the **Upload File** dialog box.
7. A confirmation dialog box displays, asking you to confirm the overwriting of the existing file. Click **Yes** to continue.
8. After uploading the updated data source file, another dialog box displays, asking you if you want to synchronize the tests with the updated data source. Click **Yes** to synchronize immediately, or click **No** if you want to synchronize later.



Note: After uploading an updated data source file, you must synchronize the data source so that associated tests are updated.

Deleting Data Sources



Note: Data sources that are being used by tests can not be deleted.

To delete a data source:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Data Sources** tab. The **Data Sources** page displays, listing all of the data sources that have been created for the system.
3. Click the **Delete** icon in the **Actions** column of the data source that you want to delete.
4. A confirmation dialog box displays. Click **Yes** to remove the data source, or click **No** to abort the deletion.

Data Sources Configuration Page

Project:<Project Name> > Project Settings > Data Sources

The **Data Sources** page lists all data sources that are configured for the integration of data-driven tests into Silk Central. Click **New Data Source** to create a new data source. For each data source, the page displays the following columns:

| Column | Description |
|------------------------|---|
| Name | The name of the data source as it displays in the Silk Central GUI and in reports. Click the name to modify the data source settings. |
| Type | Data source type: <ul style="list-style-type: none"> • CSV • JDBC • MS Excel |
| File Name | The name of the file that is used as data source, for example <code>datasource.xlsx</code> . |
| Source Control Profile | The name of the source control profile that hosts the data file. |
| Created On | Date when the data source was created. |
| Created By | User who created the data source. |
| Changed On | Date when the data source was last modified. |
| Changed By | User who last modified the data source. |
| Actions | You can perform the following actions on a data source: <ul style="list-style-type: none"> • Delete • Download • Upload • Synchronize |

Status Reasons

Status reasons qualify the status of a manual test run. This allows you to specify why a specific test run has the status **Passed**, **Failed** or **Not Executed**. You can define up to five reasons for each status. Per default, the reason **Unsupported** is added to the status **Passed**, and the reason **Unresolved** is added to the status **Failed**. Adding additional reasons for each status helps you qualify the status of a test run, for example, you could define the following reasons for the statuses:

-  **Passed:** Unsupported
-  **Failed:** Unresolved, System crashed, Bug, Help not available, Client system is unresponsive
-  **Not Executed:** On hold - internally, On hold - customer, Test environment unavailable

Defining Status Reasons

You need to have **Manage assets** permission to add or update status reasons, and **Delete assets** permission to remove status reasons. See *Tests and Executions Permissions* for detailed information.

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Status Reasons** tab.
3. Click **Add Status Reason** below the status for which you want to add a new reason. A new status reason is added to the list of reasons.
4. Click into the new status reason field and enter a descriptive text.
5. Press **Enter** to save your status reason text.

Existing status reason texts can also be modified by clicking into the status reason field and entering a descriptive text, followed by pressing **Enter**.

Project Settings Page

Project:<Project Name> > Project Settings > Project Settings

The **Project Settings** page lists the following high-level details about the active project:

| Item | Description |
|--------------------------------------|---|
| Build information file name | Build information files contain project information, including build number, build log location, error log location, and build location. Enter the name of the active project's build information file in this field. All test executions will read the build information from this specified file. |
| Project release date | Scheduled release date of the active project in the format MM/DD/YYYY. |
| File extensions to ignore in results | Result file types or other file types that should not be saved as results for test executions. |

Project Templates

The Agile project template is used to support the interaction between Silk Central and VersionOne or other Agile project management tools. The template is a project with the specific attributes sprint and release. When you create a new project based on this template, a default test container and a default folder are also created. When the project management tool creates a test, the test is added to the default folder. No source control profile and no product are specified for the test container and the folder, therefore they are marked as incomplete.

Integrations

Silk Central integrates with many different tools for many different purposes.

Code Analysis Tools Integration

DevPartner Code Coverage Integration

Silk Central integrates with the **Code Analysis** component of DevPartner Studio. You can perform code analysis on .NET applications using this integration.

For more information, see *.NET Code Analysis*.

Issue Tracking Profile Integrations

Issue tracking profiles enable Silk Central to integrate with external issue tracking systems.

Additional issue tracking systems can be configured by installing a custom plug-in. For additional information, see *Issue Tracking Integration*. Defining issue tracking profiles allows you to link tests within the **Tests** area to issues in third-party issue-tracking systems. Linked issue states are updated periodically from the third-party issue tracking system.

Atlassian Jira Software

This section describes how to configure Atlassian Jira Software (Jira) issue tracking profiles to integrate with Silk Central.

The Jira plug-in relies on the Silk Central Java API for integration, using Jira's REST API.

Versions that are currently supported for integration with Silk Central:

- Atlassian Jira Software 6, 7, 8. Versions 6 and 7 are supported but not tested.
- Atlassian Jira Software Cloud. Not supported are *next-gen* projects and projects residing in a cloud instance containing *next-gen* projects.

If you assign an external Jira issue to a test, you can enter the issue ID either with or without the project key. For example: PROJECT-13 or just 13.

Silk Central supports the following Jira field types:

- **Text Field (< 255 characters) / Free Text Field (unlimited text)**
- **Select List (single choice) / Select List (multiple choices)**
- **Date Picker / Date Time Picker**



Note: These types are provided as text fields and need to be formatted like in Jira: **Date Picker:** dd/MMM/YY (for example 24/Mar/16), **Date Time Picker:** dd/MMM/YY h:mm a (for example 24/Mar/16 4:11 AM)

- **Group Picker (single group) / Group Picker (multiple groups)**
- **Project Picker (single project)**
- **User Picker (single user) / User Picker (multiple users)**
- **Version Picker (single version) / Version Picker (multiple versions)**

Other Jira field types are rendered as single-line text fields in Silk Central.

The following Jira system fields are not supported: Attachment

In the dialogs to create issues, required custom fields and required standard fields are marked with an asterisk (*).

Adding Atlassian Jira Software Issue Tracking Profiles

To add a Jira issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **Atlassian Jira Software** from the **Type** list.

6. Type a valid **Username** and **Password**. The user name needs to be a valid Jira user profile name. These credentials will be used to access your Jira issue tracking system.



Important: Use the credentials of a Jira account with sufficient permissions, as missing permissions in Jira can cause inconsistencies when the plug-in attempts to add or update issues through this account. The Jira account requires the following permissions for each project that is used in Silk Central:

- Browse Projects
- Create Issues
- Assignable User
- Modify Reporter

Additionally, the Jira account needs to be able to access (log in to) your Jira application.

Use the following credentials to integrate with Jira cloud:

- **Username:** A Jira user profile name.
- **Password:** A Jira API token. For more information on how to generate the API token, refer to the Jira documentation.

Use the following credentials to integrate with an on-premise Jira instance:

- **Username:** A Jira user profile name.
- **Password:** A Jira password.

7. In the **URL** field, type the host name of your Jira server and the port that is used to connect to the server.
8. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
9. Optional: Select **true** from the **Show all fields** list to display all available Jira fields in the issue dialog, otherwise only the required fields are displayed. If true is selected, the Jira user defined for the issue tracking profile has to have Jira administration permissions.
10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing Atlassian Jira Software Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing Atlassian Jira Software issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.

5. *Optional*: Select a new **Type** for the issue tracking profile from the list box.
6. Type a valid **Username** and **Password**. The user name needs to be a valid Jira user profile name. These credentials will be used to access your Jira issue tracking system.



Important: Use the credentials of a Jira account with sufficient permissions, as missing permissions in Jira can cause inconsistencies when the plug-in attempts to add or update issues through this account. The Jira account requires the following permissions for each project that is used in Silk Central:

- Browse Projects
- Create Issues
- Assignable User
- Modify Reporter

Additionally, the Jira account needs to be able to access (log in to) your Jira application.

Use the following credentials to integrate with Jira cloud:

- **Username:** A Jira user profile name.
- **Password:** A Jira API token. For more information on how to generate the API token, refer to the Jira documentation.

Use the following credentials to integrate with an on-premise Jira instance:

- **Username:** A Jira user profile name.
- **Password:** A Jira password.

7. In the **URL** field, edit the hostname of your Jira server and the port that is used to connect to the server.
8. To change the Jira project, click **Load Project** to load all projects from the server and update the **Project** list box, then select a project from the **Project** list box.
9. *Optional*: Select **true** from the **Show all fields** list to display all available Jira fields in the issue dialog, otherwise only the required fields are displayed. If true is selected, the Jira user defined for the issue tracking profile has to have Jira administration permissions.
10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Bugzilla

This section describes how to configure Bugzilla issue tracking profiles to integrate with Silk Central.

The Bugzilla plug-in relies on the Silk Central Java API for integration. Silk Central communicates with Bugzilla through the XML-RPC Bugzilla Web-service introduced with Bugzilla 3.0 by using the Redstone XML-RPC library. To enable Bugzilla integration, ensure that you have installed a SOAP::Lite package for the web server. For example, for the Debian distribution install the `libsoap-lite-perl` package.

Versions that are currently supported for integration with Silk Central: Bugzilla 4.4.13, 5.0.4

Adding Bugzilla Issue Tracking Profiles

To add a Bugzilla issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.

2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **Bugzilla** from the **Type** list box.
6. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
7. Enter the URL of your Bugzilla installation. For example, `http://bugzillaserver/bugzilla`.
8. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

9. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing Bugzilla Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing Bugzilla issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Edit the **Username** and **Password**.
These credentials are used to access your issue tracking system.
7. Edit the **URL** of your Bugzilla installation.
8. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

9. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

IBM Rational ClearQuest

This section describes how to configure IBM Rational ClearQuest issue tracking profiles to integrate with Silk Central.

IBM Rational ClearQuest products provide flexible defect/change tracking and automated workflow support. The two key products are IBM Rational ClearQuest (ClearQuest) and IBM Rational ClearQuest

MultiSite (ClearQuest Multisite). To work with ClearQuest profiles, you must have the ClearQuest client software installed on the computer where the Silk Central front-end server is running. For detailed information about installing ClearQuest, refer to the ClearQuest documentation.

Versions that are currently supported for integration with Silk Central: IBM Rational ClearQuest 8.0

Adding ClearQuest Issue Tracking Profiles

To add a ClearQuest issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **IBM Rational ClearQuest** from the **Type** list box.
6. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
7. Enter the **Repository Info** of your ClearQuest installation.
This is the database name that is defined in the ClearQuest client software. The name of the sample database is `SAMPL`.



Note: It is not possible to use more than one ClearQuest schema repository. Silk Central will connect to the default ClearQuest repository.

8. Specify the **Record Type**, which is the issue type of ClearQuest.
When entering an issue in Silk Central, ClearQuest will save the issue with the issue type you define in this field. The default issue type is `Defect`.
9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

10. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing ClearQuest Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing ClearQuest issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.

6. Edit the **Username** and **Password**.

These credentials are used to access your issue tracking system.

7. Edit the **Repository Info** of your ClearQuest installation.

This is the database name that is defined in the ClearQuest client software. The name of the sample database is `SAMPL`.



Note: It is not possible to use more than one ClearQuest schema repository. Silk Central will connect to the default ClearQuest repository.

8. Change the **Record Type**, which is the issue type of ClearQuest.

When entering an issue in Silk Central, ClearQuest saves the issue with the issue type you define in this field.

9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

10. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.

- Click **Yes** to proceed with the related *Mapping Issue States* procedure.
- Click **No** to map issue states later.

Issue Manager

Issue Manager, the issue-tracking tool of Silk Central, is fully integrated with Silk Central, enabling you to correlate issues with system requirements and executed tests.

Test issues can be added and managed in the menu through **Tests > Details View > Issues**. For more information, see the Issue Manager topics in this Help.

Adding Issue Manager Issue Tracking Profiles

To add an Issue Manager issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.

4. Type a **Description** for the new profile.
5. Select **Issue Manager** from the **Type** list box.
6. Check the **Local Repository** check box to use the integrated Issue Manager profile that is part of your Silk Central instance. To connect to an external Issue Manager system, uncheck this option, in which case you will need to provide valid login credentials to access the system.
7. Type the **Issue Manager URL** of your Issue Manager installation. This is the URL you use to login to Issue Manager, though without the `login` extension at the end. Leave the default URL if you selected the **Local Repository** option.

For example, if your standard Issue Manager URL is `http://IssueManager/login`, then the correct service URL is `http://IssueManager`.

8. *Optional:* If Issue Manager should communicate with the front-end server through a different URL than the one that users use to access the Web user interface, type this URL into the **Web Service URL** field. This is required for example when users access the Web user interface through a proxy, while Issue Manager needs the internal URL or IP address of the front-end server.

9. Proceed as follows:

1. Click **Load Projects**. This action will populate the **Project** list box with all initialized Issue Manager projects to which the specified user has access to. Note that only those projects display for which Issue Manager user groups have been defined, and the defined user is a member of at least one user group.
2. Select the **Project** where Issue Manager issues are maintained.

 **Caution:** We recommend not to use identical projects for Issue Manager and Silk Central, as this limits flexibility in working with both tools on different future projects.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.

- Click **Yes** to proceed with the related *Mapping Issue States* procedure.
- Click **No** to map issue states later.

Editing Issue Manager Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing Issue Manager issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Check the **Local Repository** check box to use the integrated Issue Manager profile that is part of your Silk Central instance. To connect to an external Issue Manager system, uncheck this option, in which case you will need to provide valid login credentials to access the system.
7. Edit the **Issue Manager URL** of your Issue Manager installation if the location has changed.
This is the URL you use to login to Issue Manager, though without the `login` extension at the end. For example, if your standard Issue Manager URL is `http://IssueManager/login`, then the correct service URL would be `http://IssueManager`.
8. *Optional:* If Issue Manager should communicate with the front-end server through a different URL than the one that users use to access the Web user interface, type this URL into the **Web Service URL** field. This is required for example when users access the Web user interface through a proxy, while Issue Manager needs the internal URL or IP address of the front-end server.
9. To change the Issue Manager project, click **Load Projects** to load all projects from the server and update the **Project** list box, then select a project from the **Project** list box.
10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

11. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.

- Click **Yes** to proceed with the related *Mapping Issue States* procedure.
- Click **No** to map issue states later.

SAP Solution Manager Integration

To purchase the SAP Solution Manager Plug-In, please contact your sales representative.

SAP Solution Manager facilitates technical support for distributed systems, covering solution deployment, operation, and continuous improvement. SAP Solution Manager is a centralized, robust application management and administration solution, that combines tools, content, and direct access to SAP to increase the reliability of solutions and lower total cost of ownership. For additional information on the SAP Solution Manager, refer to the SAP Solution Manager documentation.

The integration between Silk Central and SAP Solution Manager allows for the synchronization of SAP Solution Manager business processes and incidents as Silk Central requirements and issues.

System Requirements for SAP Solution Manager Integration

The following are the system requirements for the SAP Solution Manager integration plugin:

- SAP Solution Manager 7.2.
- Silk Central 19.0 or later.
- The *Silk Central Integration Hub for SAP Solution Manager* must be installed in SAP Solution Manager.

Configuring SAP Solution Manager

Configure SAP Solution Manager to enable the integration with Silk Central.

For additional information, see

[Configuration_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#) and [User_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#), which are included in the SAP Solution Manager plug-in.

For this integration to get valid links from SAP Solution Manager to Silk Central, the front-end server machine of Silk Central must have a valid fully qualified name.

Adding SAP Solution Manager Issue Tracking Profiles

To add a SAP Solution Manager issue tracking profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Issue Tracking** tab. The **Issue Tracking** page opens, listing all of the issue tracking profiles that have been created for the system.
3. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Type a **Description** for the new profile.
6. Select **SAP Solution Manager** from the **Type** list box.
7. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
8. Type the **URL** of the Z-SCTM_ADAPTER web service of SAP Solution Manager.
9. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
10. Click **Load Default Incident Type** to load all possible incident types and populate the **Default Incident Type** list box.

11. Select an incident type from the **Default Incident Type** list box. If you are creating links to existing incidents in SAP Solution Manager and you do not specify the incident type, this type will be used for it.
12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

13. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Enabling Requirements Integration with SAP Solution Manager

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **SAP Solution Manager** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.
8. Enter the **URL** of the **Z_SCTM_ADAPTER** web service of SAP Solution Manager.
9. Enter the **Username** and **Password**.
10. Click **Test Connection** to confirm that the host and user credentials are correct. If the settings are correct, a **Test connection was successful** message displays.
11. Click **Load** to load all available solution and branch combinations and select the correct one.

Silk Central stores such combinations of solutions and branches as projects.
12. *Optional:* Check the **Enable creation of unassigned requirements** check box to enable creation and editing of unmapped requirements in Silk Central projects that are configured for integration.
13. Click **OK** to confirm and close the dialog.

Microsoft Team Foundation Server

This section describes how to configure Team Foundation Server (TFS) issue tracking profiles to integrate with Silk Central.

Versions that are currently supported for integration with Silk Central: Team Foundation Server 2015, 2017

Installing a Team Foundation Server Web Service Proxy

To communicate with a TFS, you need to install a Team Foundation Server Web Service Proxy as an interface to the TFS.

To install a TFS proxy service:

1. Install IIS and the Web Deployment Tool, for example with the Web Platform Installer from <http://www.microsoft.com/web/downloads/platform.aspx>.

2. Open the Windows Features dialog and ensure that ASP .NET 4.x is checked.

In Microsoft Windows 10, the feature is located under:

```
Internet Information Services/World Wide Web Services/Application  
Development Features
```

3. Download the Team Foundation Server Web Service Proxy from **Help > Tools**.
4. Unzip the downloaded package.
5. Open a command shell and type `DotNetTfsWebServiceProxy.deploy.cmd /Y` to install the proxy service. If an error message box displays, stating that ASP.NET 4 is required, refer to <http://msdn.microsoft.com/en-us/library/k6h9cz8h.aspx>.
6. Open the IIS Manager.
7. Verify that the new website exists.
8. In the root folder of the virtual directory, open the `Web.config` file.
9. In the `appSettings` section of the file, modify the value of the key `WorkItemTrackingCacheRoot` to a local directory.

For example:

```
<appSettings>  
  <add key="WorkItemTrackingCacheRoot" value="C:\temp" />  
</appSettings>
```

10. If the directory you have specified does not exist, create it.

11. Grant **Everyone** full rights on the specified directory.

For additional information on deploying an ASP.NET Web Application, refer to <http://msdn.microsoft.com/en-us/library/ee942158.aspx>.



Note: If you are upgrading Silk Central from version 16.5 or earlier, you need to re-deploy the TFS proxy service. Delete the existing TFS proxy service, then continue with step 3 above. When you are done, restart IIS.

Adding TFS Issue Tracking Profiles

You need to configure a Team Foundation Server Web Service Proxy, which is provided by Micro Focus to enable Silk Central to access TFS issues.



Important: This is not the Team Foundation Server Proxy provided by Microsoft.

To add a TFS issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **Team Foundation Server** from the **Type** list box.
6. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
7. Type the **Domain** of the TFS user.
8. Enter the URL of your TFS installation.
For example, `http://tfsserver:8080/tfs`.
9. Type the **Collection** to which your project belongs.
For example `DefaultCollection`.
All projects in the collection are listed in the **Projects** list box.

10. Select the **Project** from the list box.
11. Type the URL of your TFS proxy. For example `http://tfsproxyserver/DotNetTfsWebServiceProxy_deploy/TfsWebServiceProxy.asmx`.
12. Click **Load Work Item Type**. The **Work Item Type** list box is populated with the available work item types.
13. Select the **Work Item Type** from the list box.
14. Click **Load Initial State**. The **Initial State** list box is populated with the states that are allowed for the selected work item type.
15. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

16. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing TFS Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing TFS issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Edit the **Username** and **Password**.

These credentials are used to access your issue tracking system.
7. Edit the **Domain** of the TFS user.
8. Edit the **URL** and **Collection** of your TFS installation.
9. Select a different **Project**.
10. Edit the URL and port of your TFS proxy.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

StarTeam Issue Tracking Profiles

This section describes how to configure StarTeam issue tracking profiles to integrate with Silk Central.

StarTeam is a software change management and configuration management tool that enables coordination and management of the software delivery process.

To work with StarTeam profiles and use the go-to-link functionality for change requests in StarTeam, you must have the StarTeam Cross-Platform Client software installed on the computer where the browser is running.

Versions that are currently supported for integration with Silk Central: StarTeam 15.x, 16.x, 17.0

Adding StarTeam Issue Tracking Profiles

To add a StarTeam issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Type a **Description** for the new profile.
5. Select **StarTeam** from the **Type** list box
6. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
7. Type the **Hostname** of your StarTeam server and the **Port** that is used to connect to the server. If this setting has not been changed, use the default port 49201.
8. Specify the type of **Encryption** that the profile supports.
9. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
10. Click **Load View** to load all views for the selected project and populate the **View** list box, then select a view from the **View** list box.
11. Select the issue type from the **Item Type** list.



Note: Selecting an item type other than **ChangeRequest** only works if this type has a StarTeam workflow configured.

12. Click **Load Status Field** to load all enumeration fields for change requests and populate the **Status Field** list box, then select a status field from the **Status Field** list box.
If you are using a custom workflow in StarTeam, this field is the workflow driver field in StarTeam that maps to the Silk Central issue state.
13. Select a type of link from the **Link Type** list box.

starteam:// External ID links on the **Issues** tab will open the cross platform client.

http:// External ID links on the **Issues** tab will open the issue in the StarTeam web UI.

14. If you selected `http://` in the **Link Type** list box, enter the web server address of the StarTeam web UI in the **WebServer** field.
15. Select **Yes** or **No** in the **Workflow** field. Select **Yes** to show the required fields that are specified in the workflow of the selected **View**.
16. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

17. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Editing StarTeam Issue Tracking Profiles

When the server or login credentials of your issue tracking system change, you must edit your issue tracking profile accordingly.

To edit an existing StarTeam issue tracking profile:

1. In the menu, click **Issues > Issue Tracking Integrations**. The **Issue Tracking** page displays, listing all issue tracking profiles that have been created for the system.
2. Click on the name of the issue tracking profile that you want to modify. The **Edit Issue Tracking Profile** dialog box opens.
3. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
4. Edit the **Description** of the profile.
5. *Optional:* Select a new **Type** for the issue tracking profile from the list box.
6. Edit the **Username** and **Password**.
These credentials are used to access your issue tracking system.
7. Edit the **Hostname** of your StarTeam server and the **Port** that is used to connect to the server.
8. Modify the type of **Encryption** that the profile supports.
9. To change the StarTeam project, click **Load Project** to load all projects from the server and update the **Project** list box, then select a project from the **Project** list box.
10. To change the view, click **Load View** to load all views for the selected project and populate the **View** list box, then select a view from the **View** list box.
11. Select the issue type from the **Item Type** list.



Note: Selecting an item type other than **ChangeRequest** only works if this type has a StarTeam workflow configured.

12. To change the workflow driver field, click **Load Status Field** to load all enumeration fields for change requests and populate the **Status Field** list box, then select a status field from the **Status Field** list box.
If you are using a custom workflow in StarTeam, this field is the workflow driver field in StarTeam that maps to the Silk Central issue state.
13. Select a type of link from the **Link Type** list box.

starteam:// External ID links on the **Issues** tab will open the cross platform client.

http:// External ID links on the **Issues** tab will open the issue in the StarTeam web UI.

14. If you selected `http://` in the **Link Type** list box, enter the web server address of the StarTeam web UI in the **WebServer** field.
15. Select **Yes** or **No** in the **Workflow** field. Select **Yes** to show the required fields that are specified in the workflow of the selected **View**.
16. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

17. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.
 - Click **Yes** to proceed with the related *Mapping Issue States* procedure.
 - Click **No** to map issue states later.

Requirements Management Tools Integration

 **Important:** Before upgrading from Silk Central 15.0 or earlier to the current version, perform a **Synchronize All** to synchronize all your requirements with your integrated requirements management tools.

Integrating external requirements management enables you to coordinate the requirements management features of Silk Central with other tools you may already be working with. Integration is configured on the **Requirements Management** page in **Project:<Project Name> > Project Settings**.

 **Note:** You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

Silk Central supports integration with external requirements management systems (RMS) through its open interface. Creating a plug-in and integrating it into Silk Central allows integrating any RMS. Refer to the *Silk Central API Help* for information about the interfaces that enable proper integration of external RMS.

Synchronizing Requirements

Enabling synchronization of requirements between Silk Central and an external RMS enables Silk Central to receive changes that occur in the external RMS whenever a synchronization is executed. If a project has external RMS integration enabled, the master system for requirements is automatically the external system. This means that synchronization is always from the external RMS tool to Silk Central. Requirements can no longer be edited in Silk Central.

Property mapping functionality allows you to map property fields between Silk Central and external requirement tools. For example, a custom field in Silk Central called `User` might be equivalent to a property field in Caliber called `Field_2`. The property mapping feature ensures that changes to requirement-property fields are accurately refreshed between projects. If you don't use the property mapping feature, only the name and the description of the requirements are mapped. For more information, see *Editing Property Mapping*.

Requirements can be synchronized in one of several ways:

| | |
|--|--|
| Manual synchronization | Click Synchronize Changes at the root folder level on the Properties page to synchronize only requirements that have changed since the last synchronization. |
| Forced manual synchronization | Click Synchronize All at the root folder level on the Properties page to force synchronization of all requirements. |
| Automatic scheduled synchronization | Based on globally defined Silk Central schedules. |
| Automatic online synchronization | Changes to requirements are automatically propagated between tools. This option is available for Caliber only. It requires Caliber client installation on the application server and MPX enabled on the Caliber Server (refer to the Caliber Installation Guide for detailed information on enabling MPX). To enable automatic synchronization between Caliber and Silk Central, add the file <code>ss.jar</code> to the application server at <code>Program Files\Silk\Silk Central <version>\lib</code> and restart the application server. Contact customer care to obtain the file. Requirement data is automatically updated in Silk Central when changes are made in Caliber and traces in Caliber are updated when test assignment changes are performed in Silk Central. This type of online synchronization is only available when projects are configured with the current baseline. |

Automatic synchronization of requirements between Silk Central and external requirements management tools can be configured to occur based on global schedules. For details on configuring global schedules, see the *Administration* topics in this Help.



Note: The **Open Caliber** buttons open whatever program is registered as the default program for opening files of extension `.crm`. On some machines, this may be the requirement viewer, rather than Caliber. This behavior can be changed by your administrator. The client program is called `caliberrm.exe`. When properly configured, the program opens to the requirement that is selected in Silk Central.

The binder icon on the project node of the **Requirements** tree indicates the status of RM integration for the project:

No configuration RM integration is not available.

Manual configuration Requirement import, upload, and synchronization can be done only by clicking the corresponding buttons on the project node in **Requirements** >  > **Properties**.

At the project level, the **Properties** page includes the following properties:

| Property | Description |
|-----------------------------|---|
| System | The external tool with which integration has been enabled. |
| Status | Whether or not integration has been enabled. |
| Project Name | The name of the external project that the Silk Central project is associated with. |
| Requirement Types | The requirement types that are shared between projects. |
| Last Synchronization | Date and time of the last synchronization. |
| Last Synchronization Status | The status of the last synchronization including the number of created, updated, and deleted items. |



Note: When integration between Caliber and Silk Central with automatic online synchronization has been enabled, the project node displays the current status of the online requirements change listener. The three possible statuses for such projects are: `Connected` (synchronized), `Reconnected` (synchronization recommended), and `Disconnected`.

Synchronizing Requirements Across Tools



Note: Use the property mapping feature to map property fields. If you don't use the property mapping feature, only the name and the description of the requirements are mapped. For more information, see *Editing Property Mapping*.

To synchronize requirements between Silk Central and an externally configured requirements management tool

1. In the menu, click **Requirements** > **Details View** .
2. Select the Project node in the **Requirements** tree.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **Synchronize Changes**.
5. Click **Yes** on the **Synchronize Requirements** confirmation dialog box to begin synchronization. A dialog box opens when synchronization is complete, displaying synchronization statistics, including the number of requirements that have been created, updated, and deleted.
6. Click **OK** to complete the synchronization. Any updates that were made to mapped requirements in your externally configured requirements management tool are now reflected in the **Requirements** tree.

Synchronizing Requirements based on Schedules

You can configure global schedules to automatically synchronize requirements between Silk Central and an external requirements management tool. To enable automatic synchronization between Caliber and Silk Central, add the file `ss.jar` to the application server at `Program Files\Silk\Silk Central <version>\lib` and restart the application server. To obtain the file, contact customer care.

To synchronize requirements based on globally defined schedules:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Requirements Management** tab.
3. Click **Edit Schedules**. The **Edit Schedules** dialog box opens.
4. Select a predefined global schedule from the **Synchronize changes** list to define when changes in the external system should be synchronized with Silk Central.
5. Select a predefined global schedule from the **Synchronize all** list to define when all requirements in the external system should be synchronized with Silk Central.
6. Click **OK**.

For information on creating global schedules, see [Adding Global Schedules](#).

Defining Email Notification for Automatic Synchronization Events

You can define an email notification to alert users when errors occur during automated synchronization of requirements between Silk Central and external requirements management tools. All notification recipients receive copies of the synchronization log files.

To define an email notification for automatic synchronization events:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Requirements Management** tab.
3. Click **Edit Notification**. The **Edit Notification** dialog box displays.
4. Check the **Enable notification** check box.
5. Select a user name from the **Username** list box.
6. If required, add additional email addresses for other recipients in the **Other email addresses** field. Use semicolons to separate multiple email addresses.
7. Click **OK**.

Atlassian Jira Software Integration

Silk Central offers a requirements integration with Atlassian Jira Software, which is based on the project and issue tracking system Jira. All issue types (the generic object type in Jira) can be synchronized. Since Atlassian Jira Software offers the creation of custom issue types, you can specify a list of Atlassian Jira Software issue types in Silk Central that you want to synchronize, such as for example `Bug` or `Improvement`.

Atlassian Jira Software versions that are currently supported for integration with Silk Central:

- Atlassian Jira Software 6, 7, 8. Versions 6 and 7 are supported but not tested.
- Atlassian Jira Software Cloud. Not supported are *next-gen* projects and projects residing in a cloud instance containing *next-gen* projects.

Limitations: Synchronizing Jira requirements only works for requirements of the configured Jira project. If requirements are linked to requirements (epics) which are located in a different Jira project, these requirements will not be synchronized.

Enabling Integration with Atlassian Jira Software

To enable integration with Atlassian Jira Software:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **Atlassian Jira Software** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Enter the **URL** of your Atlassian Jira Software server.
9. Type a valid **Username** and **Password**. The user name needs to be a valid Jira user profile name.
These credentials will be used to access your Jira issue tracking system.



Important: Use the credentials of a Jira account with sufficient permissions, as missing permissions in Jira can cause inconsistencies when the plug-in attempts to add or update issues through this account. The Jira account requires the following permissions for each project that is used in Silk Central:

- Browse Projects
- Create Issues
- Assignable User
- Modify Reporter

Additionally, the Jira account needs to be able to access (log in to) your Jira application.

Use the following credentials to integrate with Jira cloud:

- **Username:** A Jira user profile name.
- **Password:** A Jira API token. For more information on how to generate the API token, refer to the Jira documentation.

Use the following credentials to integrate with an on-premise Jira instance:

- **Username:** A Jira user profile name.
- **Password:** A Jira password.

10. In the **Issue Types** field, specify the issue types that you want to synchronize, separated by semicolons (;).

All issue types (the generic object type in Jira) can be synchronized. Since Atlassian Jira Software offers the creation of custom issue types, you can specify a list of Atlassian Jira Software issue types in Silk Central that you want to synchronize, such as for example `Bug` or `Improvement`.

11. *Optional:* To limit the amount of requirements that are being synchronized, you can use the **Custom JQL** field to pre-filter requirements before they are being synchronized. Create a query using Jira Query Language (JQL) within your Jira system, then copy and paste the string into the **Custom JQL** field. For more information on JQL, visit Atlassian's [Advanced Searching Documentation](#).



Tip:

- Do not include `ORDER BY` statements in your JQL query.
- The JQL string may not exceed 2000 characters.
- The JQL query is used additionally to your configured project and issue types, it does not overrule those settings.

12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct. A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

13. Click **Load**. The **Project** list is populated with all the projects in Jira that you have permissions for.
14. From the **Project** list, select the external project with which the Silk Central project is to be integrated.
15. Click **OK** to save your settings.

Atlassian Jira Software Property Mappings

The following properties have a default mapping between Jira and Silk Central:

- User stories that are in Jira's `Backlog` are reflected in Silk Central by having the `Sprint` property value set to `Backlog`. This enables you to easily search for them by creating a filter.
- The default `Priority` values of Jira's user stories are mapped to Silk Central's `Priority` property. If you are not using Jira's default `Priority` values, you can customize this mapping.

Viewing Jira Backlog Stories in Silk Central

1. In the menu, click **Requirements > Details View**.
2. Create a new filter with the following criteria:
 - **Selection Criteria:** `External Custom Attributes`
 - **Property:** `Sprint`
 - **Value:** `Backlog`
3. Click **Apply** to apply the filter immediately or **Save and apply** to save the filter for future reuse.

Mapping Jira Priority values to Silk Central's Priority

The default `Priority` values of Jira's user stories are mapped to Silk Central's `Priority` property as follows:

- `Blocker=SC_Priority_Critical`
- `Critical=SC_Priority_High`
- `Major=SC_Priority_Medium`
- `Minor=SC_Priority_Low`
- `Trivial=SC_Priority_Low`

If you are not using Jira's default `Priority` values, or if you want to customize the mapping, your Silk Central administrator should customize the settings as follows:

1. Stop the application server.
2. Open the file `propertyMappings.properties` with a text editor. This file is located in the Jira plugin-file `JIRA Agile.zip` on the application server in the `plugins` folder.
3. Search for `SC_Priority=Priority`, beneath which you can customize the mapping of Jira's `Priority` values to Silk Central's `Priority`.



Note: Silk Central `Priority` values require the prefix `SC_Priority_`.

4. Save and close the file
5. Re-start the application server.

CA Agile Central Integration

Integrating the project management tool CA Agile Central enables you to define user stories which you can then use as requirements in Silk Central. You can then create tests in Silk Central to cover these

requirements. To update the tests you have created with any changes to the corresponding user stories in CA Agile Central, synchronize the requirements.

The following user story properties are always synchronized from the CA Agile Central REST Service. Custom fields are also synchronized if they are specified in the integration configuration. You can use these properties to help filter and categorize the user stories.:

| Property | Description |
|-------------------------------------|---|
| Name | The name of the user story in CA Agile Central. |
| CA Agile Central ID | The identifier of the user story in CA Agile Central. This property is the External ID of the requirement in Silk Central. |
| Description | The description of the user story in CA Agile Central. |
| Iteration | The iteration in CA Agile Central in which the user story is included. |
| Project | The project in CA Agile Central in which the user story is included. |
| Release | The project release in which the user story is included. |
| State | The state of progress of the user story in CA Agile Central. The following states are available: <ul style="list-style-type: none">• Defined• In-Progress• Completed• Accepted |
| Plan Estimate (Story Points) | The estimated time in CA Agile Central for the user story to be completed. The time is estimated in story points. |

Enabling Integration with CA Agile Central

To enable integration with CA Agile Central:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **CA Agile Central** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.

This is the name that is displayed in lists where the profiles are available for selection.
8. Type the **URL**.
9. Type valid **Username** and **Password** credentials for CA Agile Central, or type a valid **API Key** instead. An API Key can be used to access your CA Agile Central subscription data without using your username and password. For more information about CA Agile Central API Keys, refer to the [CA Agile Central Application Manager page](#).
10. *Optional:* If direct access to the Internet is restricted, and the CA Agile Central REST service is located in a machine outside the scope of the internal network, use the **Proxy Host** and **Proxy Port** fields to specify a proxy through which Silk Central can connect to CA Agile Central.

 **Note:** To connect to CA Agile Central by using a proxy, you need to fill out both fields, **Proxy Host** and **Proxy Port**.

11. *Optional:* If you want to synchronize custom fields, type their names into the **Custom Fields** field, separated by commas. The internal field names must be used, not the display names.
12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct. A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.

 **Note:** Consult your system administrator if you are not able to establish a connection.

13. Click **OK**.
14. Click **Load**. The **Project** list box is populated with all the projects from all workspaces in CA Agile Central that you have permissions for.
15. From the **Project** list box, select the external project with which the Silk Central project is to be integrated.
16. Click **OK** to save your settings.

Caliber Integration

This section describes how to integrate Caliber with Silk Central.

Configuring integration with Caliber requires the definition of Caliber login credentials. Whenever requirements are synchronized between Silk Central and Caliber, these credentials are used to login to Caliber, thus checking out a Caliber license. The license is set free as soon as the synchronization process has completed. We recommend creating a dedicated Caliber user for synchronization purposes, which should be used by all Silk Central requirements integrations with Caliber. This ensures that only a single Caliber license is used for the process of synchronization.

Caliber versions that are currently supported for integration with Silk Central: Caliber 11.4, 11.5

 **Note:** Caliber user defined attribute **Multiple selection user list** and **Multiple selection group list** must be mapped to the Silk Central text attribute type and not list type.

Enabling Integration with Caliber

To enable integration with Caliber:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.

 **Note:** You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **Caliber** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Type the **Hostname** of the machine where the external server is installed.
9. Type valid **Username** and **Password** credentials for the requirements management server.
10. Click **Test Connection** to confirm that the host and user credentials you have entered are correct. A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

11. Click **OK**.

12. From the **Project** list box, select the external project with which the Silk Central project is to be integrated. The requirement types that are available with the selected project are automatically populated into the **Requirement Types** field. The baselines that are available with the selected project are automatically populated into the **Baseline** field.

13. Select a **Baseline** from the external project that should be integrated with the Silk Central project. Your selections are displayed on the **Edit Configuration** dialog box.

14. Click **OK**.

15. Select one or more requirement types from the external project that should be integrated with the Silk Central project.

Use **Ctrl+Click** to select multiple requirement types.

16. Click **OK** to save your settings.

Baseline Support for Caliber Integration

You can select the current baseline or existing user-defined baselines for Caliber integration. You cannot import modified baselined requirements into Silk Central. Requirements that are not of the current baseline can only be changed in Caliber if the version of the requirement that is used for the baseline is changed. Such changes are only updated within Silk Central requirements when a manual or scheduled synchronization is performed.

You can change a baseline after you import it into Silk Central. You can change the configured baseline to a different user-defined baseline or the current baseline. After such a change, the next synchronization of the baseline, either manual or scheduled, will update the Silk Central project and update, create, or delete requirements as required. When a baseline is changed, a message displays that states that the changes will take effect after the next synchronization.

Handling Test Assignments in Caliber

Assigned tests are displayed, managed, and created as traces ("Trace to") of synchronized requirements in Caliber.

The External Traceability for Silk Central must be enabled for a Caliber project within the Caliber Administrator and the correct Silk Central front-end server must be configured. Click **Edit** for the Silk Central External Traceability for correct project within the Caliber Administrator.

When a project is synchronized with a Silk Central project, the assigned tests of the synchronized Silk Central requirements will display as traces ("Trace To") of the Caliber requirement. If you edit these assignments in Silk Central, the changes will be reflected immediately in Caliber.

Copying Caliber-Integrated Projects

To manage Caliber baselines when copying Silk Central projects:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Requirements Management** tab.
3. Verify that the baseline you want to save is selected.



Note: If a baseline is changed, you must perform a synchronization to update the project requirements with the baseline changes, before you can copy an associated Silk Central project. The integration configuration is only copied if you select a baseline other than the current baseline. If you select the current baseline, you need to specify if you want to keep the integration configuration in the original project or move it to the copied project.

4. If the baseline that you want to save is not selected, click **Edit Configuration**. The **Edit Configuration** dialog box displays.

5. Click **Browse** next to the **Project name** field. The **Browse Projects** dialog box opens.
6. Select the baseline you want to save, then confirm your selection.
7. In the menu, click **Projects > Project List** . The **Projects** page displays, listing all existing projects and project baselines.
8. Click  in the **Actions** column of the project you want to copy.



Note: For full details on copying a project, see the *Administration* topics in this Help.

The **Copy Project** dialog box displays.

9. Select the items you want to copy into the new project, then confirm your selection.
10. Apply the baseline that you want to continue working with to the Silk Central project.



Note: After copying a project, the original project and the copy are identical. Define on which you will continue working on by applying the corresponding baseline.

Creating a Baseline for a Caliber-Integrated Project

To create a baseline of a Caliber-integrated Silk Central project:

1. Create a baseline of the Silk Central project. A Silk Central message box displays during the process, asking if you want to enable the Caliber integration for the new baseline.
2. Click **Yes**. When the new baseline is created, the **Baseline Project - Adapt Project Settings** dialog box opens.
3. Click **Edit** to change the requirements integration settings.

The **Edit Configuration** dialog box opens.

4. Click **Browse** next to the **Project name** field. The **Browse Projects** dialog box opens.
5. Select a Caliber project, a baseline, and the requirement type for synchronization.
6. Click **OK** to close the **Browse Projects** dialog box.
7. Click **OK** to close the **Edit Configuration** dialog box.
8. Click **Finish** to close the **Baseline Project - Adapt Project Settings** dialog box.

Generating Tests from Scenarios and Simulations

Generate tests from the Caliber objects **Scenario** and **Simulation**.

To generate tests from the Caliber objects **Scenario** and **Simulation**:

1. Establish integration with Caliber.
For additional information, see *Enabling Integration with Caliber*.
2. Select the **Scenario** or **Simulation** in the **Requirements** tree from which you wish to generate tests.
3. On the **Properties** tab, click **Generate Tests**. The **Generate Tests** dialog box displays.
4. Use **Ctrl+Click** or **Shift+Click** to select the tests that you want to generate.
5. Click **Generate All** to generate all tests, or click **Generate Selected** to generate the selected tests. The **Tests** tree displays.
6. Select the folder to which you want to add the generated tests.



Note: Tests that already exist in the selected destination folder are not created, but updated. No tests in the folder are deleted.

7. Click **OK**.

IBM Rational DOORS Integration

This section describes how to integrate Silk Central and IBM Rational DOORS. Silk Central 20.6 offers out-of-the-box support for IBM Rational DOORS, without the need to customize anything in your DOORS installation. If you are upgrading from Silk Central 15.5 or earlier though, and using customized dx1 files,

you will need to overwrite the dxl files in the lib\dxl folder within your Silk Central installation folder with your customized dxl files.

IBM Rational DOORS versions that are currently supported for integration with Silk Central: IBM Rational DOORS 9.5, 9.6

Enabling Integration with IBM Rational DOORS

To integrate Silk Central and DOORS, install the DOORS client on the Silk Central front-end server machine. If you use more than one front-end server machines, you must install the DOORS client on each front-end server that you have configured for integration with DOORS.

To enable integration with DOORS:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **IBM Rational DOORS** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. In the **RM service URL** field, type the URL of the Silk Central DOORS requirement Web Service.
The default value should point to the correct location already. For example `http://MySCTMHost:19120/services/doorsrequirementsmanagement`.
9. Type valid **Username** and **Password** credentials for the requirements management server.
10. The default DOORS client installation path is displayed in the **DOORS Installation Path** field. If this path is not correct, enter the correct destination within the front-end server directory structure.
11. *Optional:* To synchronize only specific requirements, based on a filter that has been specified and activated for a view in DOORS, specify the name of the view in the **DOORS Filter View** field.
12. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.
A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: If **Test Connection** fails with the error 80004005 Unspecified error: Access denied: Doors session is not authenticated, the issue might be that the DOORS setting **Prompt to open all modules from last session on startup** is enabled. To solve this issue, disable this setting in the DOORS options. Consult your system administrator if you are not able to establish a connection.

13. Click **Browse** next to the **Project name** field to open the **Browse Requirement Types** dialog box. From the **Project** field, select the external project with which the Silk Central project is to be synchronized.
The requirement types that are available with the selected project are automatically populated into the **Requirement types** field. Select the requirement types that are to be synchronized and click **OK**. Use **Ctrl+Click** to select multiple requirement types.
Your selections are now displayed on the **Edit Configuration** dialog box.
14. Click **OK** to save your settings.



Caution: As the DOORS application object is used for communication, and this object does not support login data, but rather requires a running DOORS client, Silk Central starts each DOORS client process with the provided login data and then uses that same data for all subsequent application objects. Therefore only one set of DOORS login credentials is supported for communication at one time. It is recommended that you use the same DOORS credentials for all configurations so that integration tasks can be performed on the front-end server for all projects at the same time. When a second set of credentials is used, the second set only works after all sessions using of the first set of credentials have timed out.

IBM Rational DOORS Next Generation Integration

This section describes how to integrate Silk Central and IBM Rational DOORS Next Generation. Silk Central 20.6 offers out-of-the-box support for IBM Rational DOORS Next Generation, without the need to customize anything in your DOORS installation.



Note: All artifacts of the project are synchronized, except the DOORS collection artifacts **Module** and **Collection** themselves.

IBM Rational DOORS Next Generation versions that are currently supported for integration with Silk Central: IBM Rational DOORS Next Generation 6.0

Enabling Integration with IBM Rational DOORS Next Generation

To enable integration with IBM Rational DOORS Next Generation:

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings**.
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **IBM Rational DOORS Next Generation** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. In the **DOORS Next Generation URL** field, type the fully qualified URL of the IBM Rational DOORS Next Generation server, for example `https://MyDOORSHost.MyDomain.com:9443/rm`.
9. In the **Authentication URL** field, type the fully qualified URL of the IBM Rational Jazz Team Server, for example `https://MyDOORSHost.MyDomain.com:9443/jts`.



Note: The new IBM Rational Authentication Server is not supported for authentication, only the IBM Rational Jazz Team Server.

10. Type valid **Username** and **Password** credentials for the IBM Rational DOORS Next Generation server.
11. Click **Test Connection** to confirm that the host and user credentials you have entered are correct.
A **Connection successful** message box displays if the settings are correct. Click **OK** to continue.



Note: Consult your system administrator if you are not able to establish a connection.

12. Click **Load** next to the **Project name** list to populate the list with the available DOORS projects. Select the external project with which the Silk Central project is to be synchronized.

13. Click **OK** to save your settings.

SAP Solution Manager Integration

To purchase the SAP Solution Manager Plug-In, please contact your sales representative.

SAP Solution Manager facilitates technical support for distributed systems, covering solution deployment, operation, and continuous improvement. SAP Solution Manager is a centralized, robust application management and administration solution, that combines tools, content, and direct access to SAP to increase the reliability of solutions and lower total cost of ownership. For additional information on the SAP Solution Manager, refer to the SAP Solution Manager documentation.

The integration between Silk Central and SAP Solution Manager allows for the synchronization of SAP Solution Manager business processes and incidents as Silk Central requirements and issues.

System Requirements for SAP Solution Manager Integration

The following are the system requirements for the SAP Solution Manager integration plugin:

- SAP Solution Manager 7.2.
- Silk Central 19.0 or later.
- The *Silk Central Integration Hub for SAP Solution Manager* must be installed in SAP Solution Manager.

Configuring SAP Solution Manager

Configure SAP Solution Manager to enable the integration with Silk Central.

For additional information, see

[Configuration_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#) and [User_Guide_SilkCentral_Integration_Hub_for_SAP.pdf](#), which are included in the SAP Solution Manager plug-in.

For this integration to get valid links from SAP Solution Manager to Silk Central, the front-end server machine of Silk Central must have a valid fully qualified name.

Adding SAP Solution Manager Issue Tracking Profiles

To add a SAP Solution Manager issue tracking profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Issue Tracking** tab. The **Issue Tracking** page opens, listing all of the issue tracking profiles that have been created for the system.
3. Click **New Profile** to open the **New Issue Tracking Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Type a **Description** for the new profile.
6. Select **SAP Solution Manager** from the **Type** list box.
7. Type a valid **Username** and **Password**.
These credentials will be used to access the issue tracking system.
8. Type the **URL** of the Z-SCTM_ADAPTER web service of SAP Solution Manager.
9. Click **Load Project** to load all projects from the server and populate the **Project** list box, then select a project from the **Project** list box.
10. Click **Load Default Incident Type** to load all possible incident types and populate the **Default Incident Type** list box.
11. Select an incident type from the **Default Incident Type** list box. If you are creating links to existing incidents in SAP Solution Manager and you do not specify the incident type, this type will be used for it.

12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

13. If the trial connection is successful, a confirmation dialog box displays, asking you if you want to map internal issue states to the states of the newly defined profile.

- Click **Yes** to proceed with the related *Mapping Issue States* procedure.
- Click **No** to map issue states later.

Enabling Requirements Integration with SAP Solution Manager

1. In the menu, click **Projects > Project List**.
2. Select the project to which you want to establish integration.
3. In the menu, click **Project:<Project Name> > Project Settings** .
4. Click the **Requirements Management** tab.
5. Click **New Integration**.



Note: You can enable creating, editing and deleting requirements through Silk Central, rather than only allowing modifications through the integrated tool. You can allow modifications directly in Silk Central's UI, and/or through Silk Central's web services by checking the respective **Allow requirement create, edit, and delete...** check boxes.

The **New Integration** dialog box opens.

6. Select **SAP Solution Manager** from the list and click **Next**. The **Edit Configuration** dialog box opens.
7. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
8. Enter the **URL** of the `Z_SCTM_ADAPTER` web service of SAP Solution Manager.
9. Enter the **Username** and **Password**.
10. Click **Test Connection** to confirm that the host and user credentials are correct. If the settings are correct, a **Test connection was successful** message displays.
11. Click **Load** to load all available solution and branch combinations and select the correct one.
Silk Central stores such combinations of solutions and branches as projects.
12. *Optional:* Check the **Enable creation of unassigned requirements** check box to enable creation and editing of unmapped requirements in Silk Central projects that are configured for integration.
13. Click **OK** to confirm and close the dialog.

Working with External Properties

This section explains how to work with external properties in Silk Central.

Editing External Properties

To edit external properties:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement for which you intend to edit external properties.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **Edit External Properties**.

The **Edit External Properties** dialog box displays. All properties of the external requirement are displayed here. Edit all properties as required.



Note: Editable properties on this dialog box offer input fields and controls with which you can edit the properties. If a mapping rule exists for an attribute, the attribute will be tagged with a trailing asterisk (*).

5. Click **OK** to save your changes and close the dialog box.

Viewing External Properties

To view external properties:

1. In the menu, click **Requirements > Details View** .
2. Select the requirement.
3. Click the **Properties** tab. The **Properties** page displays the properties of the selected tree-element.
4. Click **View External Properties**. The **View External Properties** dialog box displays. All properties of the external requirement are displayed here.
5. Close the dialog box.

Editing Property Mapping

The property mapping functionality allows you to map property fields between Silk Central and external requirements management tools. For example, a custom requirement property in Silk Central called `User` might be equivalent to a custom property in Caliber called `User_ID`. The property mapping feature ensures that requirement property fields are accurately populated between projects during requirement uploading and importing. If there are multiple requirement types, you must map each type separately.



Note: If you don't use the property mapping feature, only the name and the description of the requirements are mapped.

To edit property mapping:



Note: The following applies for synchronizing properties with boolean values: since Silk Central does not support the boolean data type, you need to create a single select list with the following two list items:

- Name = `Yes`, Numeric Weight = 1
- Name = `No`, Numeric Weight = 0



Note: When you map lists, requirement properties from both products need to have the same named options for each list item, and they are case sensitive.

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Edit Property Mapping** for the configured external tool. The **Edit Property Mapping** dialog box opens.
5. Select an external requirement type from the **Requirement types** list. All custom requirements of that type are then displayed below in the selection box.
6. Select the custom requirement property for which you are establishing mapping.
7. From the list box on the right, select the Silk Central custom property to establish mapping to the external custom property you have selected.
8. Click **Add Mapping** to map the requirements. The results are displayed in the **Custom property mapping** box.
9. The **System property mapping** box displays the two pre-configured mappings for requirement name and description, which cannot be removed.
10. Click **OK** to save your changes.

Deleting Property-Mapping Value Pairs

To delete a property-mapping value pair:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Edit Property Mapping** for the configured external tool. The **Edit Property Mapping** dialog box opens.
5. Select the property-mapping value pair in the **Custom property mapping** select box.
6. Click **Remove Mapping**.
7. Click **OK** to save your changes.

Disabling Requirements Management Integration

To disable requirements management integration configuration:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click the **Disable Configuration** button of the requirements management tool for which you want to disable integration.

All integration data and functionality is disabled, but not deleted from the project.

Removing Requirements Management Integration

To remove requirements management integration:

1. Select the project.
2. In the menu, click **Project:<Project Name> > Project Settings** .
3. Click the **Requirements Management** tab.
4. Click **Remove Configuration** of the requirements management tool for which you want to remove integration.
This button is only enabled if the configuration is disabled.
The **Remove External Association** dialog box displays.
5. Click **Yes**.

All related data is removed from the database.

Source Control Profile Integrations

AccuRev

This section describes how to configure AccuRev source control profiles.

AccuRev is a software configuration management tool which addresses complex parallel and distributed development environments with its stream-based architecture, to accelerate your development process and improve asset reuse.

AccuRev versions that are currently supported for integration with Silk Central: AccuRev 7.2

Adding AccuRev Source Control Profiles

To create a AccuRev source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **AccuRev** from the **Source control system** list box.
6. Type a valid AccuRev **Username** and **Password**, if required.
7. Type the **AccuRev server** name of the AccuRev server you want to access. If you are not using AccuRev's default port, type <server name>:<port number>.
8. Type the name of the AccuRev **Stream**, for example `backing stream`.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing AccuRev Source Control Profiles

To modify a AccuRev source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the **Username** and **Password**.
These credentials are used to access your repository.
6. Edit the **AccuRev server** name of the AccuRev server you want to access. If you are not using AccuRev's default port, type <server name>:<port number>.
7. Edit the name of the AccuRev **Stream**, for example `backing stream`.
8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running

parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Edit the **Project path** you want this profile to use.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Apache Commons Virtual File System

This section describes how to configure Virtual File System (VFS) source control profiles.

A VFS is an abstraction layer on top of a more concrete file system. The purpose of a VFS is to allow client applications to access different types of concrete file systems in a uniform way. Apache Commons VFS provides a single API for accessing various file systems. It presents a uniform view of the files from various sources. The protocols that are currently supported for VFS by Silk Central are:

Protocol Description

| | |
|-------------|---|
| http | Copies the given file. This protocol type is also supported for copying and unpacking ZIP, JAR, or other zipped files. It is required to specify a .zip file on a http server. For example, <code>zip:http://myTestServer/myTests.zip</code> . The .zip file will be extracted on the execution server. |
| ftp | Copies the given file. This protocol type is also supported for copying and unpacking ZIP, JAR, or other zipped files. |
| smb | Server Message Block (smb) copies all files and folders. This protocol can be used instead of a UNC profile. For example, the VFS smb path <code>smb://server-name/shared-resource-path</code> is equivalent to the UNC path <code>\\server-name\shared-resource-path</code> . |



Note: When you create a new ProcessExecutor test that uses VFS for source control, you need to specify the complete path to the executable in the **Executable Name** field.

Adding VFS Source Control Profiles

To create a VFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **VFS** from the **Source control system** list box.
6. Type the URL of the VFS **Repository** you want to access. This field is case sensitive.

Specify the appropriate protocol type in the URL:

| | |
|-------------|---|
| FTP | <code>ftp://<FTP server URL></code> |
| HTTP | <code>http://<HTTP server URL></code> |
| SMB | <code>smb://<Samba server URL></code> |



Note: HTTP, FTP and SMB are also supported for zipped files. In order to point to a zipped file the URL must be adjusted to `<zipped file type>:<protocol>://<server URL pointing`

to zipped file> to include the type of the zipped file. For example, `zip:http://193.80.200.135/<path>/archive.zip` or `jar:http://193.80.200.135/<path>/archive.jar`.

7. Type a valid VFS Username and Password.

These credentials will be used to access your VFS repository. The SMB protocol allows including the domain name in the username in the following form: `domain/username`.

8. Optional: Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Type the Project path you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

10. Click OK.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing VFS Source Control Profiles

To modify a VFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the URL of the VFS **Repository** you want to access.

 **Note:** This field is case sensitive.

6. Edit the Username and Password.

These credentials are used to access your repository.

7. Optional: Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, `C:\TempSources\`.

 **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

8. Edit the Project path you want this profile to use.

9. Click OK.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Git

This section describes how to configure Git source control profiles.

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed. Every Git working directory is a full-fledged repository with complete history and full version tracking capabilities, not dependent on network access or a central server.

Adding Git Source Control Profiles

To create a Git source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **Git** from the **Source control system** list box.
6. Type the URL of the Git **Repository** you want to access.



Note: This field is case sensitive.

7. Type a valid Git **Username** and **Password**, if required.
8. *Optional:* Specify a **Branch** to check out a specific branch. Leave this field empty to check out the master branch.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.



Tip: If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan. See *Creating Execution Plans* for detailed information. This will overwrite the specified **Branch** in the Git source control profile.

Editing Git Source Control Profiles

To modify a Git source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Type the URL of the Git **Repository** you want to access.



Note: This field is case sensitive.

6. Edit the **Username** and **Password**.
These credentials are used to access your repository.
7. *Optional:* Specify a **Branch** to check out a specific branch. Leave this field empty to check out the master branch.
8. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.



Tip: If you want to check out a specific branch of your repository for a specific execution plan, specify the branch name in the **Source Control Label** field of the respective execution plan. See *Creating Execution Plans* for detailed information. This will overwrite the specified **Branch** in the Git source control profile.

Microsoft Team Foundation Server

This section describes how to configure Team Foundation Server (TFS) source control profiles.

TFS versions that are currently supported for integration with Silk Central: Team Foundation Server 2015, 2017



Note: To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).

Adding TFS Source Control Profiles

To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).



Note: If your TFS project has a Git repository configured, you have to create a Git source control profile instead of a TFS source control profile. Use . as the user name and a token that you have generated in TFS as the password for the Git profile. For additional information on creating a Git source control profile, see [Adding Git Source Control Profiles](#).

To create a TFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **Team Foundation Server** from the **Source control system** list box.
6. Enter the **URL** of your TFS installation.
For example, `http://tfsserver:8080/tfs`.
7. Type the **Domain** of the TFS user.
8. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing TFS Source Control Profiles

To add a TFS source control profile, the Microsoft Visual Studio Team Explorer Everywhere 2015 command-line client (English version) needs to be installed on the front-end server and every execution server on which you want to use the source control profile. You can download the CLC from the [Microsoft GitHub repository](#). You must run Team Explorer Everywhere and accept the license agreement to make the integration work. For more information, refer to this [knowledge base article](#).

To modify a TFS source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the URL of the TFS **Repository** you want to access.
6. Edit the **Username** and **Password**.
These credentials are used to access your repository.
7. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

8. Edit the **Project path** you want this profile to use.
9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Silk Test Workbench

This section describes how to configure Silk Test Workbench source control profiles.

Silk Test Workbench is an automated testing tool that accelerates the functional testing of complex applications. Silk Test Workbench provides support for testing applications developed in a wide variety of development tools including Java, .NET, browser-based web applications, and COM components, including both ActiveX controls and automation objects. With Silk Test Workbench, you can record user sessions with your applications to create tests, enhance the test by adding validation and test logic, and play back tests to ensure that the applications work as expected.



Note: To add a Silk Test Workbench source control profile, the Silk Test Workbench client needs to be installed on every execution server on which you want to use the source control profile.

Adding Silk Test Workbench Source Control Profiles

To create a Silk Test Workbench source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **Silk Test Workbench Test** from the **Source control system** list box.
6. Select a database type from the **Database type** list box, either SQL Server or Oracle.



Note: The Microsoft Office Access database type is not supported. When connecting to a Silk Test Workbench Oracle database, there is a Silk Test Workbench requirement that the TNS name must be the same as the host server name.

7. Enter the name of the Silk Test Workbench database server in the **Database server** field.
8. Enter the name of the Silk Test Workbench database in the **Database name** field.
9. Enter the port number of the Silk Test Workbench database server in the **Database port** field.
10. Enter the Silk Test Workbench database schema or owner name in the **Database schema/owner name** field.
11. Enter the name of the database user in the **Database user** field.
12. Enter the password for the database user in the **Database password** field.
13. *Optional:* Enter the name and password of a valid Silk Test Workbench user in the **Silk Test Workbench User name** and **Silk Test Workbench password** fields. If these fields are left empty, the Windows user that runs the execution server service is used.
14. Click **Retrieve**. All projects are listed in the **Projects** list.
15. Select one or more projects.
16. Click **OK**.

Editing Silk Test Workbench Source Control Profiles

To modify a Silk Test Workbench source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Edit the Silk Test Workbench **Database type**.



Note: The Microsoft Office Access database type is not supported.

6. Edit the other settings as required.
7. Click **OK**.

StarTeam

This section describes how to configure StarTeam source control profiles.

StarTeam promotes team communication and collaboration through centralized control of all project assets. Protected yet flexible access ensures that team members can work whenever and wherever they like through an extensive choice of Web, desktop, IDE, and command-line clients. StarTeam offers a uniquely comprehensive solution that includes integrated requirements management, change management, defect tracking, file versioning, threaded discussions, and project and task management.

StarTeam versions that are currently supported for integration with Silk Central: StarTeam 15.x, 16.x, 17.0

Adding StarTeam Source Control Profiles

To create a StarTeam source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.

3. Click **New Profile** to open the **New Source Control Profile** dialog box.
 4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
 5. Select **StarTeam** from the **Source control system** list box.
 6. Type the **Hostname** of your StarTeam server.
 7. Type the **Port** that is to be used to connect to the StarTeam server.
If the port is not changed, use the default port 49201.
 8. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
 9. Specify if the profile supports **Encryption**.
 10. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.
The working folder must be a local path. For example, C:\TempSources\.
-  **Warning:** It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir.
11. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
 12. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.

 **Note:** If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing StarTeam Source Control Profiles

To modify a StarTeam source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Choose from the following options:
 - Edit the **Hostname** of your StarTeam server.
 - Edit the port that is to be used to connect to the StarTeam server. If the port is not changed, use the default port 49201.
 - Specify if the profile supports **Encryption**.
6. Edit the **Project path** you want this profile to use.
7. Edit the **Username** and **Password**.
These credentials are used to access your repository.
8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.
The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Subversion

This section describes how to configure Subversion (SVN) source control profiles.

If you set a Source Control Label on the execution plan, you can check out specific Subversion branches or tags. Branches and tags are encoded in the Subversion URL. For example, `http://MyHost/svn/MyApp/trunk` is a Subversion URL and `http://MyHost/svn/MyApp/tags/build1012` is a Subversion tag.

To check out branches and tags in Silk Central, the main repository URL that is specified in the source control profile must contain the `trunk` directory. The `trunk` directory in the Subversion URL is then replaced with the label that is specified for the execution plan.

Adding Subversion Source Control Profiles

To create a Subversion source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
 2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
 3. Click **New Profile** to open the **New Source Control Profile** dialog box.
 4. Type a **Name** for the new profile.
- This is the name that is displayed in lists where the profiles are available for selection.
5. Choose **Subversion** from the **Source control system** list box.
 6. Type the URL of the Subversion **Repository** you want to access.

If you do not know the URL of the repository, consult your Subversion administrator.



Note: To use SSH, add `ssh` to the URL, for example `svn+ssh://<hostname>:<port>`.

7. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
8. Type a valid **SSH username** and **SSH password** or **SSH keyfile**.
These credentials are used to access the SSH server. The password overrides the keyfile, so if you only have a keyfile, leave the **SSH password** field empty. If you use a keyfile, the path to the keyfile must be valid on every execution server that uses the source control profile.
9. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

10. Type the **Project path** you want this profile to use. Click **Browse** to display the **Select Project Path** dialog box. The **Select Project Path** dialog box opens. Select the desired project path in the tree view and click **OK**. Leaving this field empty sets the project path to the root directory.
11. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing Subversion Source Control Profiles

To modify a Subversion source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.

This is the name that is displayed in lists where profiles are available for selection.

5. Edit the **URL** of the Subversion repository you want to access.

If you do not know the URL of the repository, please consult your Subversion administrator.



Note: If you cannot check out files, when editing the URL of the Subversion **Repository**, delete the source control mirrors directory on your execution server. For example, `C:\ProgramData\SilkCentral\SrcCtrlMirrors`.



Note: To use SSH, add `ssh` to the URL, for example `svn+ssh://<hostname>:<port>`.

6. Edit the **Username** and **Password**.

These credentials are used to access your repository.

7. Type a valid **SSH username** and **SSH password** or **SSH keyfile**.

These credentials are used to access the SSH server. The password overrides the keyfile, so if you only have a keyfile, leave the **SSH password** field empty. If you use a keyfile, the path to the keyfile must be valid on every execution server that uses the source control profile.

8. *Optional:* Edit the **Common working folder** to which the Silk Central execution server copies the source files.

The working folder must be a local path. For example, `C:\TempSources\`.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter `#sctm_source_root_dir`.

9. Edit the **Project path** you want this profile to use.

10. Click **OK**.

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Silk TestPartner

This section describes how to configure TestPartner source control profiles.

TestPartner is a Micro Focus product that tests Web- and Microsoft Windows-based applications that use Microsoft technologies. TestPartner records user actions to quickly produce powerful tests. Each recorded test displays as a series of actions in clear, concise steps that can be easily understood by all testers, from novice to expert. You can record user sessions with the application, add validation functions, and replay the sessions later to ensure that the application works as expected.



Note: To add a TestPartner source control profile, the TestPartner client needs to be installed on every execution server on which you want to use the source control profile.

Adding TestPartner Source Control Profiles

To create a TestPartner source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .
2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click **New Profile** to open the **New Source Control Profile** dialog box.
4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
5. Select **TestPartner** from the **Source control system** list box.



Note: When connecting to a TestPartner Oracle database, there is a TestPartner requirement that the TNS name must be the same as the host server name.

6. Select a database type from the **Database type** list box, either SQL Server or Oracle.
7. Type the name of the database server in the **Database server** field.
8. Type the name of the database in the **Database name** field.
9. Type the port number of the database server in the **Database port** field.
10. Type the database schema or owner name in the **Database schema/owner name** field.
11. Type the name of the database user in the **Database user** field.
12. Type the password for the database user in the **Database password** field.
13. Type a valid **Username** and **Password**.
These credentials will be used to access your repository.
14. Use the **Project path** field to filter which scripts from the database are available to use as tests. Click **Browse** to display the **Select Project Path** dialog box.

The **Select Project Path** dialog box provides a tree of three levels that you can choose from:

- First level** The entire TestPartner database. Note that if you select this level, the **Project path** field will remain blank.
- Second level** The TestPartner project.
- Third level** The script type within a specified TestPartner project.

Select one of the preceding options and click **OK**. The path will be added to the **Project Path** field.

15. Click **OK**.

Editing TestPartner Source Control Profiles

To modify a TestPartner source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings** .

2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile** dialog box opens.
4. Edit the **Name** of the profile.
This is the name that is displayed in lists where profiles are available for selection.
5. Choose from the following options:
 - Edit the TestPartner **Database type**.
 - Edit the TestPartner **Database server**.
 - Edit the TestPartner **Database name**.
 - Edit the TestPartner number of the TestPartner database server.
 - Edit the TestPartner **Database schema/owner name**.
 - Edit the TestPartner **Database user**.
 - Edit the TestPartner **Database password**.
6. Edit the **Username** and **Password**.
These credentials are used to access your repository.
7. Edit the **Project path** you want this profile to use.
8. Click **OK**.

Universal Naming Convention

This section describes how to configure Universal Naming Convention (UNC) source control profiles.

Short for Universal Naming Convention or Uniform Naming Convention, UNC is a PC format for specifying the location of resources on a local area network (LAN). UNC uses the following format: `\\server-name\shared-resource-pathname`.

For example, to access the file `test.txt` in the directory `examples` on the shared server `silo`, you would write: `\\silo\examples\test.txt`.

You can also use UNC to identify shared peripheral devices, such as printers. The idea behind UNC is to provide a format so that each shared resource can be identified with a unique address.

UNC is only supported on Microsoft Windows operating systems. If you plan to use a non-Windows execution server you can use the Apache Commons VFS source control profile instead.

Adding UNC Source Control Profiles

To create a UNC source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings**.
 2. Click the **Source Control** tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.
 3. Click **New Profile** to open the **New Source Control Profile** dialog box.
 4. Type a **Name** for the new profile.
This is the name that is displayed in lists where the profiles are available for selection.
 5. Select **UNC** from the **Source control system** list box.
 6. Type the **UNC path** that you want to access.
This is the path to the location where your test sources are located.
-  **Tip:** To avoid checking out unnecessary files, specify the folder at a hierarchy level where the sources reside that are actually required for test execution.
7. *Optional:* Type the **Common working folder** to which the Silk Central execution server should copy the source files.

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir. If you do require a static working folder, make sure that it is not located within the specified **UNC path**, as this will lead to recursive checkouts.

8. Type a valid **UNC username and **UNC password**.**

These credentials are required to access the UNC path of the configuration file.

9. Click **OK.**

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Editing UNC Source Control Profiles

To modify a UNC source control profile:

1. In the menu, click **Project:<Project Name> > Project Settings .**

2. Click the **Source Control tab. The **Source Control** page opens, listing all of the source control profiles that have been created for the system.**

3. Click on the name of the source control profile that you want to modify. The **Edit Source Control Profile dialog box opens.**

4. Edit the **Name of the profile.**

This is the name that is displayed in lists where profiles are available for selection.

5. Type the **UNC path that you want to access.**

This is the path to the location where your test sources are located.



Tip: To avoid checking out unnecessary files, specify the folder at a hierarchy level where the sources reside that are actually required for test execution.

6. *Optional:* Type the **Common working folder to which the Silk Central execution server should copy the source files.**

The working folder must be a local path. For example, C:\TempSources\.



Warning: It is recommended to leave this field empty and let Silk Central use a temporary working folder instead. Specifying a static working folder can lead to unexpected results when running parallel executions. The temporary location can be queried from the predefined parameter #sctm_source_root_dir. If you do require a static working folder, make sure that it is not located within the specified **UNC path**, as this will lead to recursive checkouts.

7. Edit the **UNC username and **UNC password**.**

These credentials are required to access your configuration file's UNC path.

8. Click **OK.**

Silk Central attempts a trial connection to the external system using the information you have provided.



Note: If an error occurs, please review the information that you have supplied, or consult your administrator.

If the trial connection is successful, you are returned to the **Source Control** page.

Test Automation Tools Integration

Unified Functional Testing (UFT) Tests

Unified Functional Testing (UFT) software, formerly known as HP QuickTest Professional (QTP), provides functional and regression test automation for software applications and environments.

Creating or Editing Unified Functional Testing (UFT) Tests

To execute UFT tests, the Unified Functional Testing software must be installed on the same computer on which Silk Central execution servers are installed.

To create or edit a UFT test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Unified Functional Testing Test** from the **Type** list box. Click **Next**.

The **Unified Functional Testing Test Properties** dialog box appears.

6. In the **Test** field, select the source control location of the UFT test to be executed.



Tip: You can select the test (.tsp) itself, or the folder where the test is located.

7. *Optional:* Check the **Validate UFT add-ins** option to validate all specified add-ins of the selected UFT test.
8. *Optional:* Check the **Show UFT application** option to display the application under test during test execution.
9. *Optional:* Check the **Log debug messages** option to log debug messages in the test run results.
10. *Optional:* Use the **Inherit from parent** check boxes to inherit the settings from the parent UFT test.
11. Click **Finish**.



Note: Parameters are passed to the UFT test by defining parameters with the same name as the test parameters that are defined in the UFT test itself.

The original UFT result files, `run_results.xml` and `run_results.html`, are available in the **Files** tab of the test run results.

JUnit Tests

JUnit is a unit testing framework for the Java programming language. Use this test type for JUnit, Silk4J, Selenium, and UFT Developer tests.

Creating or Editing JUnit Tests

To create or edit a JUnit test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **JUnit Test** from the **Type** list box. Click **Next**.

The **JUnit Test Properties** dialog box appears.

6. Specify a valid Java **Classpath** to use on the execution server.

We recommend to use a relative classpath. The relative classpath is then expanded to the full classpath on the execution server. By using a relative classpath, changes on the location of the source control profile do not require additional changes to the classpath.

The relative classpath must point to the root node of the test container containing the JUnit test, for example `JUnit_tests`. The relative classpath on the execution server is then expanded to include the working folder of the source control profile, for example `C:\temp`, and the test file names, for example `tests.jar`. For example, if your tests are packaged in a jar file called `tests.jar` and additional jar files that are needed for the execution are stored in a sub folder `lib`, your classpath would be `./tests.jar;./lib/*`.

You can also use a fully qualified classpath. The fully qualified classpath must point to the archive or folder in which the test classes reside.

7. *Optional:* In the **Test class** field, type the fully qualified name of the JUnit test class. If the field is left blank, all tests in the classpath will be executed.

8. *Optional:* In the **Test method** field, type the name of the appropriate test method.

The method must be available in the test class. If the **Test method** field is left blank, all tests that are included in the suite will be executed.

9. *Optional:* Set the **Java home directory** to the installation path of the Java Runtime Environment (JRE). The path must be valid on the execution server on which the test runs.



Note: JUnit tests can be executed in the latest version of JRE 1.8. If you use an older JRE version, messages containing `java.lang.UnsupportedClassVersionError` or `Unrecognized option: -javaagent` will display in the **Messages** tab.

10. *Optional:* In the **JVM options** field, you can specify the command-line options. You can specify multiple options, but you have to type them in the right order. For example, to use the client VM and set the maximum size of the heap to 512MB, type `-client -Xmx512m`. You can also use environment variables, for example `-Xmx${MY_MEMORY}`.

11. Check the **Record external AUT coverage only** check box to get only the code coverage for the application under test that is defined for the execution plan in the **Code Analysis Settings** portion of **Silk Central > Execution Planning > Code Analysis**.

If the check box is not checked, then the code coverage from the virtual machine that executes the test is recorded too.

12. *Optional:* In the **Test coverage path** field, type the JAR libraries or the specific class files to monitor for code coverage information. Note that the coverage path of the JUnit test is specified here, not the path of the application under test.

We recommend using the relative coverage path from the test container root node, which is then expanded on the execution server. You can also use a fully qualified path. Use semicolons to separate multiple jar files, as the following examples show:

- `tests.jar;C:\tests\testutils.jar`
- `${testenv}\tests.jar;${testenv}\testutils.jar`



Note: The coverage path setting is disregarded if the **Record external AUT Coverage** check box is checked.

13. Click **Finish**.



Note: Parameters are passed to the Java process as system properties, for example - `Dhost_under_test=10.5.2.133`. Use the `System.getProperty()` method to access the system properties. For example, to access the previously passed `host_under_test`, use `System.getProperty("host_under_test");`.

Accessing Silk Central Parameters through Java System Properties

Any JUnit test class can access a test parameter of the underlying test as a Java system property; the launcher passes these parameters to the executing machine using the "-D" VM argument.

In addition to the customer-defined parameters, you can always call the following Java system properties from a JUnit test:

| Parameter | Description |
|---|--|
| <code>#sctm_execdef_name</code> | The name of the execution plan that is currently executed. If the execution was triggered from the Tests area, the parameter has the value <code>Unassigned Tests</code> . |
| <code>#sctm_execdef_id</code> | The numeric identifier (database key) of the execution plan that is currently executed. |
| <code>#sctm_product</code> | The name of the product as defined in the executed test container. |
| <code>#sctm_version</code> | The name of the version to which the results of the execution are associated. |
| <code>#sctm_build</code> | The name of the version to which the results of the execution are associated. |
| <code>#sctm_keywords</code> | A comma separated list which contains the keywords that are defined for this execution plan. |
| <code>#sctm_regular_execdef_run_id</code> | The numeric identifier (database key) of the regular execution plan run. For setup and cleanup tests this property contains the run ID of the regular run. |
| <code>#sctm_test_results_dir</code> | The path of the directory where the result files of the test reside. |
| <code>#sctm_test_name</code> | The name of the Silk Central test being executed. |
| <code>#sctm_test_id</code> | The numeric identifier (database key) of the test being executed. |
| <code>#sctm_source_root_dir</code> | The local path of the test container's root directory, where all files under version-control reside. |
| <code>#external_id</code> | The unique identifier of a test case in a test package to be executed. You can use this as a filter if only specific test cases in a test package should be executed. |
| <code>#is_test_package_root</code> | If <code>true</code> , the test being executed is the root node of the test package (default). If <code>false</code> , only specific test cases in a test package specified by <code>#external_id</code> are executed. |
| <code>#sctm_data_driven_parent_test_name</code> | The name of the parent test, if the executed test is a data driven row. |
| <code>#sctm_data_driven_parent_test_id</code> | The numeric identifier (database key) of the parent test, if the executed test is a data driven row. |

MSTest Tests

MSTest is a command line utility from Microsoft that executes unit tests created in Visual Studio. Use this test type for MSTest, Silk4NET, and UFT Developer tests.

Creating or Editing MSTest Tests

 **Note:** The MSTest plugin supports the test type unit tests only. Results of other test types are ignored.

To create or edit an MSTest test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.

 **Note:** Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **MSTest Test** from the **Type** list box. Click **Next**.

The **MSTest Test Properties** dialog box appears.

6. You can run a test container, a test list, an ordered test, a test class, or a single test method:
 - To run a test container, click **Browse** next to the **Test file** field and select a `.dll` file. You can constrain the run by typing in just a **Test class** or both a **Test class** and a **Test method**.
 - To run a test list, click **Browse** next to the **Test file** field and select a `.vsmdi` file (Visual Studio Test Meta Data). Type a name in the **Test list name** field. You can constrain the run by typing in both a **Test class** and a **Test method**.
 - To run an ordered test, click **Browse** next to the **Test file** field and select a `.orderedtest` file.

 **Important:** To run ordered tests, leave all fields except the **Test file** field blank.

7. Click **Finish**.

 **Note:** To run MSTest on an execution server, a Visual Studio distribution or the Visual Studio Test Agent have to be installed. Add the folder that contains `MSTest.exe` to your path variable and restart the execution server. The default path is: `C:\Program Files (x86)\Microsoft Visual Studio <version>\Common7\IDE`. Currently MSTest distributed with Microsoft Visual Studio/Visual Studio Test Agent 2015 is supported.

NUnit Tests

NUnit is a unit testing framework for the Microsoft .NET programming language.

Creating or Editing NUnit Tests

 **Note:** We recommend that you add the `.\bin` folder of your NUnit installation to the system path. Click **Start > Control Panel > System > Advanced > Environment Variables** to add a path like `C:\Program Files\NUnit 2.2\bin` to the system environment variable `PATH`.

To create or edit an NUnit test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.

3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **NUnit Test** from the **Type** list. Click **Next**.

The **NUnit Test Properties** dialog box appears.

6. Click **Browse** to locate and select the NUnit assembly from which you want to pull a test.

7. Type the working directory in the **NUnit Directory** field.

For NUnit 2, this directory is the local path to the file `nunit-console.exe`, such as `C:\Program Files\NUnit 2.2\bin`. For NUnit 3, this directory is the full path to the file `nunit3-console.exe`, such as `C:\Program Files (x86)\NUnit.org\nunit-console\nunit3-console.exe`.

8. In the **NUnit Options** field, type one or more NUnit console command-line options to specify how NUnit tests are specified.

For example, to define that the `Flex4TestApp` NUnit test, located in `SilkTest.Ntf.Test.Flex`, should be executed, set the run option as follows:

For NUnit 2:

```
/run:SilkTest.Ntf.Test.Flex.Flex4TestApp
```

For NUnit 3:

```
/test:SilkTest.Ntf.Test.Flex.Flex4TestApp
```



Note: When you add multiple options, you have to separate the options by writing one option in each line in the field.

9. Click **Finish**.

ProcessExecutor Tests

The ProcessExecutor test type can be used to execute any command line.

Creating or Editing ProcessExecutor Tests

To create or edit a ProcessExecutor test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.

3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **ProcessExecutor Test** from the **Type** list box. Click **Next**. The **ProcessExecutor Test Properties** dialog box appears.

6. In the **Executable Name** field, type the fully qualified name of the executable.

7. In the **Argument List** field, type all arguments of the process executor test method.



Note: Multiple arguments must be on separate lines in the field.

8. Set the **Working Folder**. This is the folder where the executable is executed.

9. If the executed command produces JUnit test result files, for example an Ant or Maven JUnit execution, you can specify a folder pattern that is scanned for JUnit result xml files. The folder location is relative to the working directory, for example `test-results`. The root folder where the search starts is the **Working Folder**, in which all JUnit output files where the path contains the entered pattern are retrieved.

During execution of the executable the following two environment variables can be used:

| Environment Variable | Description |
|--------------------------------|--|
| SCTM_EXEC_RESULTSFOLDER | All files in this folder are saved as result files in Silk Central. If an output .xml file is created during execution in this directory, it is processed by Silk Central. |
| SCTM_EXEC_SOURCESFOLDER | This is the folder where all source files used during the execution are located. |



Note: To define these environment variables as arguments for a test, use the predefined parameters that are available for all test types, instead of using the specific environment variable that is only available for ProcessExecutor tests. For example, use `#sctm_test_results_dir` instead of `SCTM_EXEC_RESULTSFOLDER`.

Examples

To configure a ProcessExecutor test that runs the Windows Script Host from the command line with the `/c` switch and two parameters, specify the following:

- `/c` - A command that specifies that the command line should terminate after execution.
- `cscript` - The Windows Script Host.
- `parareadwrite.js` - The jsript file.

To run the test on a Linux server, you can use the following Python script:

```
#!/usr/bin/env python
#
import sys
import os
print 'command and arguments: ' + str(sys.argv)
print 'SCTM_EXEC_RESULTSFOLDER: ' +
str(os.environ.get('SCTM_EXEC_RESULTSFOLDER'))
```

Instead of using the environment variable `SCTM_EXEC_RESULTSFOLDER`, you can set the predefined parameter `#sctm_test_results_dir` as an argument in Silk Central. The argument will be replaced with the corresponding environment variable when the script is executed.

Silk Performer .NET Explorer Tests

The Silk Performer .NET Explorer test type is used to execute .NET Explorer scripts or a specific test case of a script.

Creating or Editing Silk Performer .NET Explorer Tests

To create or edit a .NET Explorer test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Performer .NET Explorer Test** from the **Type** list box. Click **Next**.

The **.NET Explorer Test Properties** dialog box opens.

6. Browse to and select the .NET Explorer script to apply to the test.

The script is a .nef file.

7. Enter the path to the `NetExplorer.exe` executable on the execution server.

For example `C:\Program Files\MyCustomSPFolder\DotNET Explorer\NetExplorer.exe`.

You can also define an environment variable on your execution servers and refer it, for example `{SP_HOME}\NetExplorer.exe`.

8. In the **Test case** field, type the name of the .NET Explorer test case to execute. If this field is left blank, all test cases within the script are executed. It is not possible to specify multiple test cases.



Note: The test cases `InitTestCase` and `EndTestCase` are always executed.

9. Click **Finish**.

Silk Performer Tests

Creating or Editing Silk Performer Tests

To create or edit a Silk Performer test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Performer Test** from the **Type** list box. Click **Next**.

The **Silk Performer Test Properties - Select Project** dialog box appears.

6. Click **Browse** to select a Silk Performer project that has been saved to your file system and click **Next**.
7. Select one of the workload profiles that has been defined for the project from the **Workload** list.
8. Click **Finish** to create the test case.

Silk Test Classic Tests

Silk Central's Silk Test Classic interface offers a reliable means of automating Silk Test Classic users. Each test case of a Silk Test Classic script executes within its own test execution and produces its own results.

The Silk Test Classic Plan test type is used for Silk Test Classic test plans. A test plan usually is a hierarchically-structured document that describes the test requirements and contains the statements, 4Test scripts, and test cases that implement the test requirements.

When you observe running Silk Test Classic executions on the **Activities** page, the currently running execution plan offers a hyperlink that opens a **Details View**. This view allows you to closely monitor the state of the currently running execution plan. For Silk Test Classic test executions, the center component of

this view consists of two parts: The upper part shows general information about the test, script, test case, and test data. The lower part shows all output messages generated by Silk Test Classic, along with their severity.

In previous versions of Silk Central, Silk Test Classic invocation was implemented through a command-line interface. The new interface works using interprocess communication. You can specify whether or not Silk Central's Silk Test Classic interface should be used by configuring test-container settings.

For all Silk Test Classic test parameters that are not defined through Silk Test Classic test properties in the Silk Central GUI, the Silk Test Classic default settings are used, for example from `partner.ini`.

Creating or Editing Silk Test Classic Tests

To create or edit a Silk Test Classic test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Test Classic Test** from the **Type** list box. To import multiple Silk Test Classic test cases at the same time, select **Silk Test Classic Multi-testcase import**. Click **Next**.

The **Test Properties - Select Test Script** dialog box appears.

6. Click **Browse** and select the test script file from either the defined Silk Test Classic project or the source control directory.

The test script, with file extension `.t` or `.g.t`, is defined relative to the test container's root node in the source control profile. For example `<Silk Test Classic project>/MyScript.t`.

7. Click **Next**.

The **Test Properties - Select Testcase** dialog box appears.



Note: If the Silk Test Classic script is a data-driven `.g.t` file, for example `SilkTestScript1.g.t`, then data sources are completely controlled within the script file and not through the data-driven properties of Silk Central. The **Data-driven** check box is checked by default when you use a data-driven script file. For more information about data-driven Silk Test Classic tests, refer to the Silk Test Classic documentation.

8. Select a test case from the available test cases in the defined script file or specify a custom test case. When the custom test case field is already populated, the Silk Test Classic test was automatically created, using the export functionality within Silk Test Classic. When you use the custom field to specify the test case, you can terminate the test case name with a parenthesis `()`. Inside the parenthesis, you can specify test data, including parameters. Specifying data in the custom field will override the values of the **Test data** property.
9. *Optional:* Type additional execution arguments into the **Test data** field.

These arguments are processed by Silk Test Classic during the execution of the test. If several arguments are passed to Silk Test Classic, they have to be separated by a comma (,). If a `String` argument is passed to Silk Test Classic, the argument must be set in quotation marks (`"`). When test data is more complex, we recommend that you use parameters in the test data, for example `{ParameterName}`. Parameters are replaced automatically within test executions.



Note: This field has a maximum length of 2000 characters.

10. Define whether the test should provide a TrueLog.



Tip: When running your tests with the Open Agent, this setting must be turned on both in Silk Central and in Silk Test, otherwise no TrueLog will be provided. Default execution mode for data-driven tests is plan-based. When you use script-based execution mode for a data driven test, change the *DataDrivenScriptMode* setting in the Silk Test Classic element of *SccExecServerBootConf.xml*. For more information, refer to the Silk Test Classic documentation.

11. If required, specify an option set file. By default, Silk Central closes all open Silk Test Classic option set files. To specify an option set file, specify the file name relative to the test container's root node in the source control profile.

12. Click **Finish** to create the Silk Test Classic test.



Note: If you possess Silk Test Classic test cases that require more than one hour to complete, adjust the time-out settings for Silk Central. Otherwise, Silk Central assumes an error has occurred and terminates the execution. For details about time-out settings, see the *Administration* topics in this Help.

Creating or Editing Silk Test Classic Plan Tests

To create or edit a Silk Test Classic plan test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Test Classic Plan** from the **Type** list box. Click **Next**.

The **Plan Properties** dialog box appears.

6. In the **Plan file** field, type the fully qualified name of the test plan file (.pln) to be executed. Click **Browse** to browse for the file.
7. In the **Project file** field, type the name of the Silk Test Classic project containing the file and environmental settings. Click **Browse** to browse for the project file.
8. In the **Option set** field, type the fully qualified name of the option set file containing environmental settings. Click **Browse** to browse for the option set file.
9. In the **Data file for attributes and queries** field, type the default path of the test plan initialization file. Click **Browse** to browse for the test plan initialization file.
10. In the **Test plan query name** field, type the fully qualified name of the saved test plan query.
11. Click **Finish**.

To maintain the structure of your Silk Test Classic plan test in Silk Central, execute a trial run. Once the trial run has completed successfully, the *output.xml* file is added to the result files of your execution plan run. This allows you to convert your test to a test package, giving you full control over the test's structure.

Silk Test Workbench Tests

Silk Test Workbench is an automated testing tool that accelerates the functional testing of complex applications. Silk Test Workbench provides support for testing applications developed in a wide variety of development tools including Java, .NET, browser-based web applications, and COM components, including both ActiveX controls and automation objects.

Prerequisites to Run STW.EXE with Silk Central

To run STW.EXE with a Silk Central execution server and to successfully connect to a Silk Test Workbench database, the following prerequisites are required:

1. Run the Silk Central execution server as a process, and not as a service.
2. Use database authentication to connect to the Silk Test Workbench database. Silk Test Workbench can use Windows authentication or database authentication when connecting to a Silk Test Workbench SQL Server database.
3. Configure the database server instance to use a static port.

If these prerequisites are not met, Silk Test Workbench might return the following error when trying to run STW.EXE with a Silk Central execution server:

Error 3: DSN, username, or password incorrect.

Creating or Editing Silk Test Workbench Tests

To create or edit a Silk Test Workbench test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk Test Workbench Test** from the **Type** list box. Click **Next**.

The **Silk Test Workbench Test Properties - Select Tests** dialog box appears.

6. Select one or more scripts from the **Silk Test Workbench Test** list box.
7. In the **Playback Options** field, type in the name of a predefined Silk Test Workbench playback option or leave the default value of `System Defaults`.



Note: A playback option must be set for a Silk Test Workbench test. The delete option on a Silk Test Workbench test will reset the playback option to the default value of `System Defaults`.

8. Click **Finish**.

Opening Silk Test Workbench in the Context of the Result File

Click the `result.stwx` file to open Silk Test Workbench in context of the result file `result.stwx`.

If Silk Test Workbench uses an Oracle database, ensure that the file `tnsnames.ora` exists on the machine on which you want to open Silk Test Workbench, and that the file includes an entry with a `net_service_name` that is the same as the `SID` that is configured in the Silk Test Workbench source control. For example:

```
myOrclSid=
(DESCRIPTION=
(AADDRESS=...))
```

```
(CONNECT_DATA=
(SID=myOrclSid))
```

1. In the menu, click **Execution Planning**.
2. Select your execution plan.
3. Click the **Runs** tab.
4. In the **Test Runs** grid at the bottom, click the **Run ID** of the test for which you want to see details. The **Test Run Results** dialog box opens.
5. Click the **Files** tab.
6. Click the `result.stwx` file to open Silk Test Workbench in context of the result file.



Note: Silk Test Workbench must be installed on the machine on which you are trying to open the file.

Viewing the Execution Result Files for a Visual Test

1. In the menu, click **Execution Planning**.
2. Select your execution plan.
3. Click the **Runs** tab.
4. In the **Last Executions** area, click the **Run ID** of the relevant execution plan to view test-execution results.
5. The **Actions** column displays all icons and links for Silk Test Workbench:
 - `detail.htm` - Silk Test Workbench detailed steps file.
 - `result.stwx` - Click to open the result in Silk Test Workbench .
 - `error.png` - Click to go to a screen shot of the last error.

Test Parameters Page - Silk Test Workbench

Tests > Details View > <Test Element> > Data Set

When passing parameters from a Silk Central test to a Silk Test Workbench test, these parameters must be defined as input parameters in the Silk Test Workbench script and they must have the same name in Silk Central as in the script. The following data types should be used:

| Silk Central | Silk Test Workbench |
|--------------|----------------------|
| String | Text |
| Number | Number (Long) |
| Number | Number (Long Long) |
| Number | Enumeration |
| Float | Number (Double) |
| Boolean | Boolean (True/False) |

Predefined parameter names in Silk Central start with a hash while Silk Test Workbench does not allow parameter names starting with a hash. To pass a predefined parameter from Silk Central to a Silk Test Workbench test, you have to create an additional parameter without the hash in the name in Silk Central and map the predefined parameter to the new parameter.

For example, to pass the predefined parameter `#sctm_test_results_dir` to a Silk Test Workbench script, you have to apply the following workaround:

1. Specify an input parameter in the Silk Test Workbench script with the name `resultsdir`.
2. Specify a parameter in Silk Central with the name `resultsdir` and the value `#{sctm_test_results_dir}`.

Silk Central will then replace the placeholder with the actual directory and pass it to the Silk Test Workbench script.

Silk TestPartner Tests

TestPartner is a Micro Focus product that tests Web- and Microsoft Windows-based applications that use Microsoft technologies.

Creating or Editing Silk TestPartner Tests

To create or edit a TestPartner test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Silk TestPartner Test** from the **Type** list box. Click **Next**.

The **Test Properties - Select Test Script** dialog box appears.

6. Click **Browse**.

The **Choose File** dialog box opens. The assets available in this dialog box are based on the **Project Path** defined in the associated TestPartner source control profile.

7. Select a script from the list and click **OK**.



Note: To add multiple TestPartner tests, see *Adding Multiple TestPartner Tests*.

8. In the **Playback Options** field, type in the name of a predefined TestPartner playback option or leave the default value of `System Defaults`.
9. Click **Finish**.

Adding Multiple TestPartner Tests

To add multiple TestPartner tests, you must first follow the steps described in *Creating Silk TestPartner Tests*.

To add multiple TestPartner tests:

1. On the **New Test** dialog box, select **TestPartner Test** from the **Type** list box.
2. Click **Next**. The **Test Properties - Select Test Script** dialog box opens.
3. Click **Browse**.

The **Choose File** dialog box opens. The assets available in this dialog box are based on the **Project Path** defined in the associated TestPartner source control profile.

4. In the **Playback Options** field, type in the name of a predefined TestPartner playback option or leave the default value of `System Defaults`.



Note: The import gives the defined playback option to all imported scripts. If you need to change the playback option for any scripts, edit the test after the import.

5. Click **Next**.

The **Test Properties - Select Scripts** dialog box opens.

6. In the **TestPartner Scripts** field, select the scripts to import by clicking **Ctrl+Click**.
7. Click **Finish**.

Windows Script Host Tests

Windows Script Host (WSH) is part of the Windows platform and creates an environment for hosting scripts. When a script is to be run at the execution server, WSH plays the role of host. It makes objects and services available for the script and provides a set of guidelines within which the script is executed. Among other things, WSH manages security and invokes the appropriate script engine.

Creating or Editing Windows Script Host (WSH) Tests

To create or edit a WSH test:

1. In the menu, click **Tests > Details View**.
2. Select a container or folder node in the **Tests** tree where you want to insert a new test, or select the test that you want to edit.
3. Click  (**New Child Test**) on the toolbar, or click  (**Edit**).

The **New Test/Edit Test** dialog box appears.

4. Type a name and description for the test.



Note: Silk Central supports HTML formatting and cutting and pasting of HTML content for **Description** fields.

5. If you are creating a new test, select **Windows Script Host Test** from the **Type** list box. Click **Next**.

The **Windows Scripting Test Properties** dialog box appears.

6. Click **Browse** and select a Windows scripting test script.
7. Specify the location of any required additional parameters in the **Switches** field.



Note: You may add other switches to be passed to the script. For more details on the switches that can be used, see [WSH Test Properties](#) and refer to the MS Scripting Host documentation.

8. Click **Finish**.

Supported Script Languages

WSH is language-independent for WSH-compliant scripting engines. Natively, the Windows platform supports Visual Basic Scripts, with file extension `.vbs`, and scripts written in the Java Script language, with file extension `.js`.

For other scripting languages, a dedicated script interpreter must be installed on the execution server. For example, if you install a Perl interpreter on an execution server, this will register a Perl scripting engine at the WSH environment for the extension `.pls`. Whenever a file with extension `.pls` is passed to the WSH tool, with the executable `cscript.exe`, it will invoke the appropriate interpreter because of the file extension. So the client of WSH, in this case the Silk Central Execution Server, does not need to know about the installation of the Perl interpreter.



Note: After installing a script interpreter, for example Active Perl, try to execute a script locally on the execution server by calling the WSH command line tool with a sample script before executing the script in Silk Central. To do so, open a command shell on the execution server and type `cscript <someScript>`, where `<someScript>` is the path to a script of your choice that is available on your execution server. This is exactly what Silk Central will call when executing a WSH test on an execution server. If the script is executed, then the scripting engine has been registered successfully.

The following scripting languages are WSH compatible:

| Scripting Language | File Extension |
|--------------------|----------------|
| Perl | .pls |
| Python | .PY, .PYW |
| REXX | .REXX |
| TCL | .tcl |

WSH Test Properties

| Property | Description |
|----------|--|
| Script | You can define any file where a script engine is registered for the script language the file contains. Script files under source control are deployed automatically to execution servers, comparable to test sources for other test types. |
| Switches | You can enter and pass the following settings to <code>cscript.exe</code> during the execution of the test: <ul style="list-style-type: none"> //B Batch mode suppresses all non-command-line console UI requests from the script. We recommended that you use this option to prevent a script from waiting for user input during unattended executions at the execution server. //U We recommend to that you use unicode for redirected I/O from the console. //T:nn Time-out, in seconds. The maximum time the script can run, by default = no limit. This option is used to prevent excessive execution of scripts. It sets a timer. When execution time exceeds the specified value, Cscript interrupts the script engine using the <code>IActiveScript::InterruptThread</code> method and terminates the process. There is a callback hook. If the time-out is invoked, the <code>OnTimeOut</code> function is called to permit cleanup. Although it is possible to create infinite loops using this feature, it is more useful than harmful. //logo Displays an execution banner at execution time that is visible at the beginning of the <code>log.txt</code> log file. This is the default setting. //nologo Prevents display of the execution banner at execution time. //D Enables active debugging. //E:engine Use the engine to execute a script. //Job:xxxx Execute a WSF job. //X Execute the script in debugger. |

Using Parameters in WSH Tests

Parameters that are defined for a WSH test automatically add a name and value pair to the command line as an additional argument and set the parameters as environment variables for the called process. This functionality allows you to access all parameters defined for your test within the WSH script.

For example a WSH test is defined with `myscript.js` as script and `//B` as switch. Additionally the test requires a parameter called `IPAddress` with the value `192.168.1.5` and another parameter called `Port` with the value `1492`. The resulting command line for the WSH execution in this example is:

```
cscript myscript.js //B IPAddress=192.168.1.5 Port=1492
```

Viewing Information Returned from WSH

To collect results of a WSH execution, the WSH script must generate a file called `output.xml` in the current working directory of the WSH test. All files residing in this directory are stored in the database and are downloadable through the list of files for the test execution. Files are excluded from storage when their extensions are defined under the file extensions to ignore in results property in the **Projects** area.

 **Note:** The current working directory is dynamically created for each WSH execution. Do not use an absolute path when creating the file. Any relative path used will correctly refer to the current working directory.

Any information that a script writes to the WSH standard output goes into the `log.txt` text file that resides in the current working directory. This file is stored in the database and can be viewed as it is included in the file list of the test execution.

The following example shows how to print log information from a script:

```
WScript.Echo "This info will be written to the log.txt file"
```

The XML structure of `output.xml` begins with an element *ResultElement* that defines an attribute named *TestItem*, which specifies the name of the *ResultElement*.

The *ResultElement* must contain an element named *ErrorCount*, optionally an element named *WarningCount*, and a list of Incident elements.

The *ErrorCount* and *WarningCount* elements must contain a positive number or zero. The *ErrorCount* and *WarningCount* of the top-level *ResultElement* are used for evaluating success conditions, which determine if a test has passed or failed. The XML file might contain additional elements that are not visible in the Silk Central GUI. The `output.xml` file is however stored in the database and is viewable as it is included in the file list of the executed test.

The *Incident* element represents an event that happened during the execution of the WSH test. *Message* and *Severity* are shown in the messages list of test executions in the Silk Central GUI. An *Incident* element must contain a *Message* and a *Severity* element.

The *Severity* element must hold one of the following values:

- Info
- Warning
- Error (or Exception)
- Failure

You can store additional information in the result file. The *ResultElement* may contain any number of sub-*ResultElements*, so information can be easily grouped. Sub-*ResultElements* make the result file easier to read. For compatibility reasons related to unit tests, JUnit and NUnit, *ResultElement* can be named *TestSuite* or *Test*.

The *ResultElement* may contain the following additional elements:

- *FailureCount*, which is treated the same way as error count
- *RunCount*, if a test is run multiple times
- *Timer*, for example for the duration of the test
- *WasSuccess*, for compatibility with NUnit result files
- *Asserts*, for compatibility with NUnit result files

The *Incident* element may contain a list of *Detail* elements.

The *Detail* element represents detailed information about an *Incident*. It must define a *TestName* element and an *Info* element. The *TestName* is used to give detailed information about where the *Incident* happened. The *Info* element holds detailed information about the *Incident*, for example a stack trace.

 **Note:** Up through Silk Central 8.1, the value of the *Message* and *Info* elements had to be URL encoded (ISO-8859-1). Since version 8.1.1, URL encoding is no longer allowed.

Sample Result File

```
<ResultElement TestItem="WshOutputTest">
  <ErrorCount>1</ErrorCount>
  <WarningCount>1</WarningCount>
```

```

<Incident>
  <Message>some unexpected result</Message>
  <Severity>Error</Severity>
  <Detail>
    <TestName>function main()</TestName>
    <Info>some additional info; eg. stacktrace</Info>
  </Detail>
</Incident>
<Incident>
  <Message>some warning message</Message>
  <Severity>Warning</Severity>
  <Detail>
    <TestName>function main()</TestName>
    <Info>some additional info; eg. stacktrace</Info>
  </Detail>
</Incident>
</ResultElement>

```

Java Script Sample

The following script was used to generate the sample result file. To try this script save it with the extension .js.

```

function dumpOutput(dumpFile)
{
  dumpFile.WriteLine("<ResultElement TestItem=\"WshOutputTest
\">");
  dumpFile.WriteLine("  <ErrorCount>1</ErrorCount>");
  dumpFile.WriteLine("  <WarningCount>1</WarningCount>");
  dumpFile.WriteLine("  <Incident>");
  dumpFile.WriteLine("    <Message>some unexpected result</
Message>");
  dumpFile.WriteLine("    <Severity>Error</Severity>");
  dumpFile.WriteLine("    <Detail>");
  dumpFile.WriteLine("      <TestName>function main()</
TestName>");
  dumpFile.WriteLine("      <Info>some additional info; eg.
stacktrace</Info>");
  dumpFile.WriteLine("    </Detail>");
  dumpFile.WriteLine("  </Incident>");
  dumpFile.WriteLine(" <Incident>");
  dumpFile.WriteLine("  <Message>some warning message</
Message>");
  dumpFile.WriteLine("    <Severity>Warning</Severity>");
  dumpFile.WriteLine("    <Detail>");
  dumpFile.WriteLine("      <TestName>function main()</
TestName>");
  dumpFile.WriteLine("      <Info>some additional info; eg.
stacktrace</Info>");
  dumpFile.WriteLine("    </Detail>");
  dumpFile.WriteLine("  </Incident>");
  dumpFile.WriteLine("</ResultElement>");
}

function main()
{
  var outFile;
  var fso;
  fso = WScript.CreateObject("Scripting.FileSystemObject");
  outFile = fso.CreateTextFile("output.xml", true, true);
  outFile.WriteLine("<?xml version=\"1.0\" encoding=\"UTF-16\"?
>");
}

```

```

    dumpOutput(outFile);
    outFile.Close();
    WScript.Echo("Test is completed");
}

main();
WScript.Quit(0);

```

Visual Basic Script Sample

The following Visual Basic script also generates the sample result file, and saves it as Output.xml. To try this script save it with the extension .vbs.

```

WScript.Echo "starting"

Dim outFile
Dim errCnt
Dim warningCnt

outFile = "output.xml"
errCnt = 1 ' retrieve that from your test results
warningCnt = 1 ' retrieve that from your test results

Set FSO = CreateObject("Scripting.FileSystemObject")
Set oTX = FSO.OpenTextFile(outFile, 2, True, -1) ' args: file,
8=append/2=overwrite, create, ASCII

oTX.WriteLine("<?xml version=""1.0"" encoding=""UTF-16""?>")
oTX.WriteLine("<ResultElement TestItem=""PerlTest"">")
oTX.WriteLine("  <ErrorCount>" & errCnt & "</ErrorCount>")
oTX.WriteLine("  <WarningCount>" & warningCnt & "</
WarningCount>")
oTX.WriteLine("    <Incident>")
oTX.WriteLine("      <Message>some unexpected result</Message>")
oTX.WriteLine("      <Severity>Error</Severity>")
oTX.WriteLine("      <Detail>")
oTX.WriteLine("        <TestName>function main()</TestName>")
oTX.WriteLine("        <Info>some additional info; eg.
stacktrace</Info>")
oTX.WriteLine("      </Detail>")
oTX.WriteLine("    </Incident>")
oTX.WriteLine("    <Incident>")
oTX.WriteLine("      <Message>some warning message</Message>")
oTX.WriteLine("      <Severity>Warning</Severity>")
oTX.WriteLine("      <Detail>")
oTX.WriteLine("        <TestName>function main()</TestName>")
oTX.WriteLine("        <Info>some additional info; eg.
stacktrace</Info>")
oTX.WriteLine("      </Detail>")
oTX.WriteLine("    </Incident>")
oTX.WriteLine("</ResultElement>")

```

Continuous Integration Server Integrations

Using the Silk Central web services, you can integrate your test executions into your continuous integration build system. If you are using Jenkins as your continuous integration (CI) server, you can download the Silk Central integration plugin directly from the Jenkins plugins page.

Communicating with an External System Over SSL

If the certificate of the host you want to connect to is self signed, you may receive the following error message:

sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target



Note: This error message does not display for valid certificates, which are certificates that are signed by Certificate Authorities.

To use a server with a self-signed certificate, you need to perform the following steps:

1. Download the SSL certificate from the server.

Use a browser to view the certificate and export it. For example, in Mozilla Firefox, navigate to **Tools > Page Info > Security > View Certificate > Details > Export**.

2. Start the key- and certificate-management tool *Keytool*.

Keytool is part of your JRE installation, and is located in your JRE installation folder, for example C:\Program Files (x86)\Silk\Silk Central <version>\instance_<instance number>_<instance name>\lib\jre64\bin. For additional information on Keytool, refer to the [Java SE Technical Documentation](#).

3. To add the certificate to the default Java keystore on the front-end server and application server, type for example the following command in Keytool:

```
keytool
  -importcert
  -file CERTIFICATE.crt
  -keystore "C:\Program Files (x86)\Silk\Silk Central <version>
\instance_<instance number>_<instance name>\lib\jre64\lib\security\cacerts"
```

You are prompted to type the password.

4. Type the default keystore password, *changeit*.
5. Restart the front-end server and the application server to reload the keystore.

Index

- .NET Explorer
 - test type 161, 639
- .NET Explorer test
 - configuring 161, 639
 - creating 161, 639
 - editing 161, 639

A

- accessing
 - audit log 493
 - context-sensitive execution reports 446
 - context-sensitive execution-plan-run reports 446
 - context-sensitive requirements reports 446
 - context-sensitive test reports 447
 - most recently used reports 431
 - tests from grid view 222
- accessing parameters
 - Silk Performer 313
- AccuRev
 - adding source control profiles 94, 571, 619
 - editing source control profiles 94, 572, 620
 - source control profiles 94, 571, 619
- action properties 386
- action-driven workflow
 - about 376
 - reason codes 378
 - state owner 377
- activating
 - builds 91, 501
 - components 88, 502
 - execution servers 479
 - individual values 510
 - platforms 92, 504
 - products 86, 90, 505, 508
 - project baselines 518
 - projects 518
 - schedules 513
 - value lists 510
- activities
 - across projects 335
 - managing 327
 - page 330
- Activities page
 - columns, displaying and hiding 327
 - columns, reordering 330
 - columns, resizing 330
 - default settings 330
 - deleting last execution runs 327
 - entering issues 327
 - filtering boolean-based values 329
 - filtering date-based values 328
 - filtering number-based values 328
 - filtering predefined list 329
 - filtering test runs 328
 - filtering text-based values 328
 - filters, removing 329
 - grouping 329
 - grouping test runs, overview 329
 - remove grouping 329
 - sorting test runs 330
- adding
 - builds 90, 500
 - call to shared steps 201
 - cloud profiles 481
 - components 88, 502
 - data source values to manual test steps 191
 - definite runs 294
 - dependent execution plans 285
 - device providers 484
 - exclusions 293
 - groups 470
 - keyword 139, 196
 - keywords 137
 - locations 474
 - manual testers 269
 - manual testers to configurations 259
 - multiple TestPartner tests 167, 645
 - parameters to configurations 257
 - parameters to tests 187
 - platforms 92, 503
 - products 85, 505
 - projects 515
 - schedules 512
 - subreports 353
 - test containers 123
 - test folders 125
 - user roles 459
 - value lists 509
 - versions 89, 507
- adding AUT host
 - Silk Test Classic 284, 316
- adding columns
 - inboxes 364
- adding components
 - business process tests 149
- adding flows
 - business process tests 149
- adding issue tracking profiles
 - Bugzilla 560, 594
 - IBM Rational ClearQuest 562, 596
 - Issue Manager 563, 597
 - Jira 558, 592
 - Rational ClearQuest 562, 596
 - StarTeam 569, 603
 - Team Foundation Server 567, 601
 - TFS 567, 601
- adding links
 - test containers 124
- adding profile
 - VMware vCloud 481
- adding project-wide notifications 396
- adding quality goals 337
- adding source control profiles
 - AccuRev 94, 571, 619

- Git 97, 575, 623
- Silk Test Workbench 101, 578, 626
- StarTeam 102, 579, 627
- subversion 103, 581, 629
- SVN 103, 581, 629
- TestPartner 105, 583, 631
- UNC 106, 584, 632
- VFS 96, 573, 621
- adding TFS issue tracking profiles
 - installing proxy service 566, 600
- Administrator
 - user roles 458
- advanced context-sensitive execution reports
 - enabling 447
- advanced context-sensitive requirement reports
 - enabling 448
- advanced context-sensitive test reports
 - enabling 448
- agile planning tools
 - integration default nodes 216
- all issues
 - list 344
- Analyst
 - user roles 458
- analyze results
 - automated tests 319
 - manual tests 319
- analyzing
 - server log files 494
 - Silk Performer test results 314
 - test runs 319
- analyzing test results
 - quick start task 34
- Apple Safari
 - information service, installing 486
- application server log
 - page 497
- applying
 - filters 59, 121, 301, 531
- archiving issues 391
- assigned execution plans
 - overview 212
- assigned executions
 - viewing 212
- assigned requirements
 - locating 209
 - page 210
 - removing 209
- assigned requirements page
 - tests 210
- assigned tests
 - printing 306
 - removing from requirements 73
- assigned tests page
 - executions 289
 - requirements 74
- assigning
 - attributes to execution plans 264
 - existing issues 341
 - keywords to configurations 258
 - keywords to execution plans 272
 - mobile devices to configurations 260
 - requirements to tests 208
 - tests from grid view to execution plans 287
 - tests to attributes 184
 - tests to execution plans 287
 - tests to execution plans manually 288
 - tests to execution plans through filter 288
- assigning manual tests
 - to manual testers 233, 239
- assigning requirements to tests
 - overview 208
- assigning testers
 - business components 150
- assigning tests
 - execution plans 287
 - to testing cycles 232, 239
- associating tests
 - requirements 72
- Atlassian Jira Software
 - backlog 45, 547, 609
 - integrating 44, 545, 607
 - issue tracking profiles 558, 592
 - priority 45, 547, 609
- attaching
 - files to requirements 70
 - files to test elements 189
 - links to requirements 71
 - links to test elements 189
- attaching files
 - requirements 70
- attaching links
 - requirements 71
- attachments
 - requirements 70
 - tests 189
- attachments page
 - requirements 72
- attribute types 39, 110, 532, 538
- attributes
 - assigning to execution plans 264
 - assigning to tests 184
 - creating 110, 531
 - deleting 112, 533
 - deleting from execution plans 264
 - deleting from tests 184
 - editing 111, 184, 264, 533
 - execution plans 263
 - overview 109, 531
 - page 112, 534
 - tests 184
- audit log
 - accessing 493
 - features 492
 - overview 492
 - page 493
 - viewing 493
- automated tests
 - analyze results 319
 - converting to manual 130
 - manual execution 297
- automatic fields
 - customizing 374
- automation servers

- integration 650
- average page-time trend
 - reports 454
- average transaction busy-time trend
 - reports 455

B

- backlog
 - Atlassian Jira Software 45, 547, 609
- baselines
 - Caliber 49, 550, 612
 - comparing with projects 525
 - creating for project 516
 - projects 516
- behavior on failure
 - business components, setting 150
- BIRT
 - adapting report templates 489
 - customizing report templates 427
 - customizing reports 427
 - data source settings 488
 - establishing database access 488
 - installing 487
- blocked
 - manual test 253
- bookmarks
 - reports 431
- BPT
 - business components, generating 130, 151
- browser
 - open on mobile device 274
 - settings 24
- bug workflow 379
- Bugzilla
 - adding issue tracking profiles 560, 594
 - editing issue tracking profiles 561, 595
 - issue tracking profiles 560, 594
- build
 - specifying for testing cycle 235, 242
- build information
 - projects 524
 - updates 524
- build information files
 - creating 524
 - updates 524
- build servers
 - integration 650
- building
 - keyword libraries 143, 200
 - libraries 143, 200
- builds
 - activating 91, 501
 - adding 90, 500
 - deactivating 91, 501
 - deleting 92, 501
 - editing 91, 501
 - managing 90, 500
 - sorting list 91, 501
 - tagging 91, 501
- bulk updates
 - issues 344

- business components
 - behavior on failure 150
 - business process testing 147
 - flows, adding 149
 - parameters, specifying 152
 - testers, assigning 150
 - versions 180, 206
- business process testing
 - about 147
 - business components, generating 130, 151
 - walkthrough 33
- business process tests
 - creating 148
 - creating flows 149
 - editing 149
 - importing, Excel 155
 - parameters, mapping 153
 - versions 180, 206

C

- CA Agile Central
 - integrating 47, 548, 610
 - integrating into 46, 547, 609
- calculated properties
 - about 37, 535
 - add-edit page 38, 536
 - creating 37, 536
- Caliber
 - baselines 49, 550, 612
 - copying integrated projects 49, 550, 612
 - creating baseline 50, 551, 613
 - handling test assignments 49, 550, 612
 - integrating 48, 549, 611
- call to shared steps
 - adding 201
 - dialog box description 202
- calling
 - test steps 201
- capturing
 - tests 302
- capturing screen images
 - manual testing window 249
- change notification
 - enabling 60, 121, 541
 - list of changes 60, 122, 541
 - overview 60, 121, 541
- change notifications
 - disabling 60, 122, 541
 - page 61, 122, 542
- changes
 - requirements 83
- changing
 - test status 320
- changing column labels
 - inboxes 364
- changing sort order
 - inboxes 364
- charts
 - displaying 426
 - overview 426
 - printing 426

- removing 426
- CI servers
 - integration 650
- cleaning up
 - results 303
 - test packages 129
- clearing
 - parameter assignments 187
- clearing flags
 - requirements 67
- cloud
 - integration 481
- cloud profiles
 - adding 481
 - deleting 483
 - editing 482
- cloud providers
 - page 484
- code analysis
 - build versions 283, 444
 - configuration 277, 438
 - configure DPAnalysis to launch your AUT 277, 438
 - details page 282, 442
 - enabling 275, 435
 - generating code-change impact reports 283, 444
 - Java examples 275, 435
 - Java options 275, 435
 - latest builds 283, 444
 - manual testing window 252
 - overview 274, 435
 - select classes for report dialog 284, 444
 - viewing coverage information for packages/
 - namespaces 284, 445
 - windows code analysis framework and devpartner
 - analytics 276, 437
- code analysis and automated testing
 - multiple AUTs 279, 440
 - one AUT 278, 439
 - overview 278, 438
- code analysis examples
 - Java 275, 435
- code coverage
 - overview 274, 435
 - reports 453
 - trend report 454
- code coverage reports
 - method coverage comparison 454
 - trend 454
- code-change impact
 - reports 453
- code-coverage information
 - viewing for packages/namespaces 284, 445
- collapsing
 - execution plans tree 305
 - tests tree 213
- columns
 - resizing in grid view 223
- combining
 - keyword 139, 196
 - keywords 137
- compare
 - test run results 319
- comparing
 - projects and baselines 525
 - versions 181, 207
- components
 - activating 88, 502
 - adding 88, 149, 502
 - deactivating 88, 502
 - deleting 89, 503
 - editing 88, 502
 - flows, adding 149
 - managing 88, 502
 - parameters, specifying 152
 - sorting list 89, 503
 - testers, assigning 150
 - versions 180, 206
- concepts
 - Silk Central 14
 - test automation 136
- configuration suite
 - configuration page 260
- configuration suites
 - copying 305
 - deleting 305
 - document view 309
 - editing 305
 - managing 262
- configuration testing
 - adding manual testers 259
 - adding parameters to configurations 257
 - assigning keywords to configurations 258
 - assigning mobile devices 260
 - creating configuration suites from execution plans 257
 - creating suite 257
 - overview 256
 - removing keywords 259
 - removing parameters from configurations 258
 - removing testers 259
- configurations page 261
- configuring
 - .NET Explorer test 161, 639
 - deployment environments 267
 - device providers 484
 - execution dependencies 285
 - JUnit tests 156, 634
 - manual test 129
 - MSTest test 158, 637
 - non-standard SSL ports for execution servers 480
 - JUnit test 159, 637
 - plan files (.pln) 164, 642
 - ProcessExecutor test 160, 638
 - SAP Solution Manager 53, 554, 565, 599, 616
 - schedules 511
 - Silk Performer test 162, 640
 - Silk Test Classic plan 164, 642
 - Silk Test Classic tests 162, 641
 - Silk Test Workbench test 165, 643
 - test properties 182
 - TestPartner tests 166, 645
 - UFT tests 155, 634
 - WSH test 167, 646
- configuring projects
 - quick start task 25

- settings 527
- consuming
 - tests 172, 214
- contents
 - tests 218
- context-sensitive execution reports
 - accessing 446
 - enabling 447
- context-sensitive execution-plan-run reports
 - accessing 446
- context-sensitive reports
 - accessing execution 446
 - accessing execution-plan run 446
 - accessing requirements 446
 - accessing test 447
 - enabling advanced execution 447
 - enabling advanced requirement 448
 - enabling advanced test 448
 - enabling execution 447
 - enabling requirement 448
 - enabling test 448
- context-sensitive requirement reports
 - enabling 448
- context-sensitive requirements reports
 - accessing 446
- context-sensitive test reports
 - accessing 447
 - enabling 448
- continuous integration
 - uploading keyword libraries 144
- continuous integration servers
 - integration 650
- converting
 - manual tests to keyword-driven tests 136
- copy project
 - dialog box 521
- copy to other projects
 - test elements 214
- copying
 - Caliber integrated projects 49, 550, 612
 - configuration suites 305
 - execution plan folders 305
 - execution plans 305
 - filters 58, 120, 300, 530
 - project baselines 518
 - projects 518
 - test elements 214
 - test packages 128
 - user roles 460
- coverage
 - direct mode 65
 - full mode 65
 - modes 65
 - requirements 65
- coverage page
 - requirements 65
- creating
 - .NET Explorer test 161, 639
 - advanced filters 57, 119, 299, 529
 - build information files 524
 - business process tests 148
 - configuration suite 257
 - configuration suites from execution plans 257
 - custom parameters 186
 - custom requirement properties 39, 537
 - custom schedules 292
 - custom step properties 109, 540
 - data-driven tests 191
 - execution plan parameters 265
 - execution plans 262
 - execution plans in grid view 222, 263
 - execution-server keywords 479
 - filters 56, 118, 298, 528
 - flows 149
 - issues 339
 - issues (Issue Manager) 346
 - JUnit tests 156, 634
 - keyword-driven tests 136
 - keywords 271
 - manual test 129
 - MSTest test 158, 637
 - NUnit test 159, 637
 - ProcessExecutor test 160, 638
 - project baselines 516
 - reports 419
 - reports (Issue Manager) 352
 - requirements 62
 - shared steps 202
 - Silk Performer test 162, 640
 - Silk Test Classic plan tests 164, 642
 - Silk Test Classic tests 162, 641
 - Silk Test Workbench test 165, 643
 - step properties 109, 540
 - test packages 126
 - TestPartner tests 166, 645
 - UFT tests 155, 634
 - WSH test 167, 646
- creating baseline
 - Caliber 50, 551, 613
- creating filters
 - flagged requirements 67
- creating reports
 - overview 418
- creating testing cycles 230, 237
- cross-browser testing
 - keyword-driven tests 143
- CSV
 - configuring data sources 115, 588
- CSV data
 - downloading from a data source 192
- current run
 - page 310
 - synchronizing 310
- current state 386
- custom attributes
 - editing 111, 533
- custom fields
 - defining 389
- custom measure trend
 - reports 456
- custom reports
 - BIRT 487
 - software prerequisites 487
 - SQL functions 420

- custom requirement properties
 - creating 39, 537
 - delete 41, 539
 - page 36, 534
- custom schedules
 - creating 292
- custom step properties
 - creating 109, 540
 - deleting 109, 540
 - editing 109, 540
 - page 109, 540
- custom tabs
 - creating 375
 - editing 375
- custom tabs and fields
 - Issue Details page 375
- customization properties
 - issues 374
- customizing report templates
 - BIRT 427
- customizing reports
 - overview 427
- cutting
 - test elements 214

D

- dashboard
 - adding a report panel 23, 425
 - adding panels 20
 - overview 19
- dashboard panels
 - introduction 20
 - issues created per tester 20
 - manual tests assigned to me 20
 - requirements coverage status 20
 - testing activity 20
 - testing cycle progress 20
- data mart
 - architecture 399
 - disable 406
 - DM_TestStatus 406
 - missing data 406
 - overview 398
 - performance 406
 - reporting 398
 - RV_ConfigSuiteFinishedStatus 417
 - RV_ConfigurationSuiteStatus 416
 - RV_EPFinishedStatusRollup 415
 - RV_ExecutionPlanStatusPerBuild 413, 414
 - RV_ExecutionPlanStatusRollup 415
 - RV_LatestFinishedTestStatus 410
 - RV_LatestTestStatus 409
 - RV_MaxFinishedTestRunID 412
 - RV_MaxTestRunID 412
 - RV_TestingCycleStatus 412
 - RV_TestStatusExtended 407
 - tables and views 406
 - troubleshooting 405
 - writing queries 400
 - wrong data 406
- data source values

- adding to manual test steps 191
- data sources
 - configuration page 117, 590
 - configuring CSV 115, 588
 - configuring Excel 115, 588
 - configuring JDBC 114, 587
 - data-driven tests 113, 586
 - deleting 116, 589
 - downloading Excel files 115, 588
 - limits 113, 586
 - page 117, 590
 - synchronizing 116, 589
 - uploading updated Excel files 116, 589
- data-driven
 - creating tests 191
 - downloading CSV data 192
 - editing properties 192
- data-driven properties
 - removing 192
- data-driven tests
 - atomic 192
 - data import considerations 193
 - data sources 113, 586
 - Excel 193
 - non-atomic 192
 - overview 190
 - test types 192
 - viewing activities 324
 - worksheet handling 193
- databases
 - BIRT report templates 488
- deactivating
 - builds 91, 501
 - components 88, 502
 - execution servers 479
 - individual values 510
 - platforms 92, 504
 - products 86, 90, 505, 508
 - project baselines 518
 - projects 518
 - schedules 513
 - value lists 510
- default workflows 379
- definite runs
 - adding 294
 - deleting 294
 - editing 294
 - schedules 294
 - scheduling 514
- delete
 - requirement properties 41, 539
- deleting
 - attributes from execution plans 264
 - attributes from tests 184
 - builds 92, 501
 - cloud profiles 483
 - components 89, 503
 - configuration suites 305
 - custom step properties 109, 540
 - data sources 116, 589
 - definite runs 294
 - device providers 487

- exclusions 293
- execution plan dependencies 286
- execution plan folders 305
- execution plan runs and result files 321
- execution plans 305
- execution servers 479
- filters 58, 120, 301, 530
- groups 473
- individual values 511
- issue tracking profiles 570
- issues 341
- keyword library 195
- keywords 137
- locations 475
- obsolete requirements 63
- platforms 93, 504
- products 86, 506
- project baselines 519
- project-wide notifications 397
- projects 519
- report templates 491
- requirement attachments 71
- result files 303
- results 303
- schedules 513
- server log files 495
- source control profiles 108, 585
- step properties 109, 540
- subreports 422
- test attachments 190
- test elements 214
- test run results 320
- value lists 510
- versions 90, 508
- deleting products
 - Issue Manager 367
- deleting profile
 - VMware vCloud 483
- deleting property mapping
 - requirements integration 55, 556, 619
- deleting quality goals 337
- deleting testing cycles 236, 243
- dependencies
 - deleting 286
 - editing 286
 - page 287
- deployment
 - page 267
- deployment environments
 - configuring 267
- details view
 - viewing issue statistics 340
- device providers
 - adding 484
 - configuring 484
 - deleting 487
 - editing 484
 - managing 484
- devpartner
 - dpanalysis.exe 280, 441
- DevPartner Code Coverage Integration 591
- disabling

- change notifications 60, 122, 541
- requirements integration 55, 557, 619
- displaying
 - charts 426
- displaying columns
 - grid view 222
- distributed execution
 - enabling 273
- document view
 - requirements 83
 - tests 221
 - viewing issue statistics 340
- documentation workflow 380
- DOORS
 - integrating 50, 52, 552, 553, 614, 615
- downloading
 - CSV data from a data source 192
 - report templates 423, 490
 - server log files 494
 - Silk Performer projects 312
 - Silk Performer test result packages 315
- downloading attachments
 - from requirements 71
 - from test steps 132, 205
 - from tests 190
- downloading Excel files
 - data sources 115, 588
- dynamic hardware provisioning
 - keywords 270

E

- edit execution server
 - dialog box 480
- editing
 - .NET Explorer test 161, 639
 - attributes 184, 264
 - builds 91, 501
 - business process tests 149
 - cloud profiles 482
 - components 88, 502
 - configuration suites 305
 - custom requirement properties 41, 539
 - custom step properties 109, 540
 - data-driven properties 192
 - definite runs 294
 - dependencies 286
 - device providers 484
 - execution plan folders 305
 - execution plans 305
 - external properties 54, 555, 617
 - filters 58, 119, 300, 529
 - individual values 509
 - issues (Issue Manager) 347
 - JUnit tests 156, 634
 - locations 474
 - manual test 129
 - manual test steps 129, 203
 - MSTest test 158, 637
 - NUnit test 159, 637
 - parameters 187
 - platform 92, 503

- ProcessExecutor test 160, 638
- products 86, 505
- project baselines 517
- project-wide notifications 396
- projects 517
- report parameters 353, 433
- report properties 432
- report properties (Issue Manager) 353
- report templates 490
- requirements 62
- schedules 512
- Silk Performer test 162, 640
- Silk Performer test properties 314
- Silk Test Classic plan tests 164, 642
- Silk Test Classic tests 162, 641
- Silk Test Workbench test 165, 643
- step properties 109, 540
- success conditions 194
- test containers 125
- test folders 125
- test mapping file 177
- TestPartner tests 166, 645
- tests 174
- UFT tests 155, 634
- user roles 459
- value lists 509
- versions 89, 508
- WSH test 167, 646
- editing issue tracking profiles
 - Bugzilla 561, 595
 - IBM Rational ClearQuest 562, 596
 - Issue Manager 564, 598
 - Jira 559, 593
 - Rational ClearQuest 562, 596
 - StarTeam 569, 604
 - Team Foundation Server 568, 602
 - TFS 568, 602
- editing manual test steps
 - Silk Central 129, 203
- editing profile
 - VMware vCloud 482
- editing property mapping
 - requirements integration 55, 556, 618
- editing source control profiles
 - AccuRev 94, 572, 620
 - Git 98, 575, 624
 - Silk Test Workbench 101, 579, 627
 - StarTeam 102, 580, 628
 - Subversion 104, 582, 630
 - SVN 104, 582, 630
 - TestPartner 106, 583, 631
 - TFS 100, 577, 625
 - UNC 107, 585, 633
 - VFS 96, 574, 622
- email notification
 - using
 - email notification 351
- email notification rules
 - defining 392
- email notifications
 - automatic requirement synchronization 43, 545, 607
 - defining 351
 - enabling 398
- embedding
 - reports 424
- enabling
 - advanced context-sensitive execution reports 447
 - advanced context-sensitive requirement reports 448
 - advanced context-sensitive test reports 448
 - basic email notification 398
 - change notification 60, 121, 541
 - code analysis 275, 435
 - context-sensitive execution reports 447
 - context-sensitive requirement reports 448
 - context-sensitive test reports 448
 - distributed execution 273
 - external executions 273
 - parallel execution 272
- enabling code analysis
 - execution plans 281, 442
- entering
 - issues (Issue Manager) 346
- Excel
 - configuring data sources 115, 588
 - downloading from data source 115, 588
 - importing requirements 78
 - test file 176
 - uploading updated files 116, 589
- Excel file
 - requirements 79
- Excel import
 - versioning tests 177
- Excel update
 - versioning tests 177
- exclusions
 - adding 293
 - deleting 293
 - editing 293
 - schedules 293, 513
- executing
 - attended Silk Performer tests 313
 - automated tests manually 297
 - individual execution plans 296
 - tests 295
- executing keyword-driven tests
 - without keyword library 145
- executing manual tests
 - manual testing 244
 - manual testing window 244
- executing Silk Test Classic tests
 - overview 316
- executing tests
 - quick start task 29
- execution
 - manual execution planning 226
 - schedules page 294
- execution dependencies
 - configuring 285
- execution environment 473
- execution plan folders
 - copying 305
 - deleting 305
 - editing 305
- execution plan properties

- document view 309
- execution plan run properties
 - parameters 185, 264
- execution plan runs
 - details 321
 - updating, external 298
- execution planning
 - adding manual testers to a testing cycle 231, 238
 - dependencies page 287
 - edit assigned testers 231, 238
 - history page 311
 - overview 16, 226
 - quick start task 29
 - reports 452
 - toolbar functions 307
- execution plans
 - adding dependent 285
 - adding manual testers 269
 - adding parameters 265
 - assigning attributes 264
 - assigning keywords 272
 - assigning tests 287
 - assigning tests from grid view 287
 - assigning tests through filter 288
 - attributes 263
 - combinations of automated and manual tests 267
 - configuring cleanup executions 267
 - configuring setup executions 267
 - copying 305
 - creating 262
 - creating configuration suites 257
 - creating in grid view 222, 263
 - deleting 305
 - deleting attributes 264
 - deleting dependencies 286
 - deleting runs and result files 321
 - dynamic hardware provisioning 270
 - editing 305
 - editing attributes 264
 - editing dependencies 286
 - enabling code analysis 281, 442
 - executing 296
 - external 297
 - locating assigned tests 289
 - managing 262
 - manually assigning tests 288
 - overriding test parameters 266
 - parameters 264
 - printing assigned tests 306
 - removing assigned tests 289
 - removing keywords 272
 - removing parameters 266
 - removing testers 269
 - run comparison reports 452
 - run details 321
 - run errors report 453
 - schedules 291
 - status calculation 263
 - status overview 453
 - tests not passed 453
 - tree 304
- execution plans tree
 - collapsing 305
 - expanding 305
 - filtering 306
- execution reports
 - run comparison 452
 - run errors 453
 - status overview 453
 - tests not passed 453
- execution server log
 - page 498
- execution servers
 - activating 479
 - assigning keywords 479
 - configuring cloud-hosted 478
 - configuring non-standard SSL ports 480
 - configuring physical 477
 - creating keywords 479
 - deactivating 479
 - deleting 479
 - hardware provisioning 476
 - keywords 476
 - setting up 476
- execution status
 - overview report 453
- executions
 - assigned tests page 289
 - current run page 310
 - deployment page 267
 - properties page 308
 - run dialog box 296
 - runs page 324
- expanding
 - execution plans tree 305
 - tests tree 213
- exporting
 - projects 519
 - shared steps objects to Excel 203
 - tests to Excel 175
- external execution plans
 - managing 297
- external executions
 - enabling 273
- External IDs
 - test packages 126
- external properties
 - editing 54, 555, 617
 - overview 54, 555, 617
 - viewing 54, 555, 618
- external results
 - managing 297
- external systems
 - communicating over SSL 651

F

- files
 - attaching to test elements 189
- filtering
 - boolean values in grid view 224
 - date-based values in grid view 224
 - execution plans tree 306
 - number-based values in grid view 224

- requirements tree 81
- test containers 216
- test folders 216
- tests in grid view 223
- text-based values in grid view 224
- values using predefined list in grid view 225
- filtering test runs
 - Activities page 328
- filters
 - applying 59, 121, 301, 531
 - assigning tests to execution plans 288
 - copying 58, 120, 300, 530
 - creating 56, 118, 298, 528
 - creating advanced 57, 119, 299, 529
 - deleting 58, 120, 301, 530
 - details 59, 120, 301, 530
 - editing 58, 119, 300, 529
 - global 56, 117, 298, 527
 - overview 56, 117, 298, 527
 - recent changes 59, 121, 301, 531
 - removing from grid view 225
 - removing specific from grid view 225
- find
 - global 18
 - in Silk Central 18
- finding
 - requirement properties 69
 - test properties 217
- finding assigned tests
 - requirements 73
- flagged requirements
 - creating filters 67
- flags
 - requirements 66
- flows
 - adding 149
 - adding, business components 149
 - business process testing 147
 - creating 149
 - parameters, specifying 153
 - versions 180, 206
- folders
 - managing 262
- front-end server log
 - page 496

G

- generating reports
 - code-change impact 283, 444
- generating tests
 - Scenario 50, 551, 613
 - Simulation 50, 551, 613
- getting started
 - exploring sample database 342
- Git
 - adding source control profiles 97, 575, 623
 - editing source control profiles 98, 575, 624
 - source control profiles 97, 575, 623
- global
 - reports 418
- global filters

- overview 56, 117, 298, 527
- global schedules
 - specifying 292
- grid view
 - accessing tests 222
 - assigning existing issues to tests 225
 - assigning tests to execution plans 287
 - assigning tests to requirements 72
 - creating execution plans 222, 263
 - displaying and hiding columns 222
 - filtering boolean values 224
 - filtering date-based values 224
 - filtering number-based values 224
 - filtering tests 223
 - filtering text-based values 224
 - filtering values using predefined list 225
 - grouping tests 223
 - multi-edit 222
 - overview 221
 - removing all filters 225
 - removing specific filters 225
 - reordering columns 223
 - resizing columns 223
 - restoring default settings 225
 - sorting tests 223
 - tests 221
- grid view filters
 - removing all 225
 - removing specific 225
- group properties
 - group information 359
 - initial issue state 360
 - issue related 359
- group security settings
 - Issue Manager 360
- group settings
 - page 473
- grouping
 - tests in grid view 223
- groups
 - adding 470
 - creating 470
 - deleting 473
 - editing 472
 - importing from LDAP 471
 - maintaining 470
 - security settings 360
 - setting up for Issue Manager 359

H

- handling test assignments
 - Caliber 49, 550, 612
- hiding columns
 - grid view 222
- history
 - requirements 83
- history page
 - execution planning 311
 - requirements 84
 - tests 182

I

- IBM Rational ClearQuest
 - adding issue tracking profiles 562, 596
 - editing issue tracking profiles 562, 596
 - issue tracking profiles 561, 595
- IBM Rational DOORS
 - integrating 50, 551, 613
- IBM Rational DOORS Next Generation
 - integrating 51, 553, 615
- implementing
 - keyword 139, 196
 - keywords 137
- importing
 - business process tests, Excel 155
 - groups from LDAP 471
 - manual tests, Excel 131
 - projects 520
 - tests from Excel 176
- importing requirements
 - Excel 78
 - Word 76
- importing tests
 - test file, Excel 176
- inbox settings
 - defining for projects 367
- inboxes
 - adding 362
 - default view 363
 - deleting 363, 395
 - editing 363
 - modifying default view 364
 - setting up 361
- information service
 - Mac, installing 486
- installing
 - BIRT 487
 - information service, Mac 486
 - Silk Central 14
 - Team Foundation Server Web Service Proxy 566, 600
- installing proxy service
 - TFS 566, 600
- integrating
 - Atlassian Jira Software 44, 545, 607
 - CA Agile Central 46, 47, 547, 548, 609, 610
 - Caliber 48, 549, 611
 - DOORS 50, 52, 552, 553, 614, 615
 - IBM Rational DOORS 50, 551, 613
 - IBM Rational DOORS Next Generation 51, 553, 615
 - task management tools 499
 - VersionOne 499
- integrating Atlassian Jira Software
 - overview 44, 545, 607
- integrating Caliber
 - overview 48, 549, 611
- integration
 - cloud profiles 481
 - SAP Solution Manager 52, 553, 565, 599, 616
- integration default nodes
 - tests 216
- intelligent assistant 390
- internal property values 38, 537
- iOS
 - information service, installing 486
- issue details
 - page (Issue Manager) 345
- Issue Details page
 - custom tabs and fields 375
 - customizing 373
- Issue Manager
 - about 341
 - adding issue tracking profiles 563, 564, 597, 598
 - generating reports 352
 - getting started 342
 - introduction 344
 - optional features 358
 - overview 344, 563, 597
 - routine administrative tasks 358
 - setting up your data 356
 - SOAP API 354
- issue reports
 - issues per component 453
- issue routing
 - default routing 368
 - rules 368
- issue state
 - initial, group 360
 - initial, user 365
- issue tracking
 - page 571
 - page (Issue Manager) 344
- issue tracking profiles
 - Atlassian Jira Software 558, 592
 - Bugzilla 560, 594
 - deleting 570
 - IBM Rational ClearQuest 561, 595
 - Jira 558, 592
 - mapping issue states 557
 - overview 557, 592
 - page 571
 - Rational ClearQuest 561, 595
 - SAP Solution Manager 53, 554, 565, 599, 616
 - StarTeam 568, 602
 - Team Foundation Server 566, 600
 - TFS 566, 600
- issue verification preferences
 - setting 350
- issues
 - about archiving 390
 - Activities page 327
 - archiving 391
 - assigning 341
 - bulk updates 344
 - creating 339
 - creating (Issue Manager) 346
 - deleting 341
 - document view 340
 - editing (Issue Manager) 347
 - entering (Issue Manager) 346
 - overview 339
 - page 340
 - show 344
 - synchronizing internal and external states 341
 - taking action on (Issue Manager) 350

- viewing statistics in details view 340
- viewing statistics in document view 340
- working with (Issue Manager) 349

issues per component reports 453

J

Java options

- code analysis 275, 435

Java system properties

- accessing Silk Central parameters 157, 636

JDBC

- configuring data sources 114, 587

Jira

- adding issue tracking profiles 558, 592
- editing issue tracking profiles 559, 593
- issue tracking profiles 558, 592

JUnit

- test type 156, 634

JUnit tests

- available Silk Central parameters 157, 636
- configuring 156, 634
- creating 156, 634
- editing 156, 634

K

keyboard functions

- execution planning tree 304
- libraries tree 200
- requirements tree 80
- tests tree 213

keyword

- adding 139, 196
- combining 139, 196
- deleting
 - keyword 139, 196
- implementing 139, 196
- managing
 - keyword 139, 196
- nesting 139, 196
- opening 139, 196
- parameters 139, 196
- replacing 139, 196
- sequence 139, 196

keyword libraries

- building 143, 200
- Selenium 143, 200
- uploading 144, 195

keyword library

- deleting 195

keyword-driven testing

- testing 135

keyword-driven test editor

- recommended keywords 147

keyword-driven testing

- advantages 135
- keyword recommendations, algorithm 147
- overview 135
- parameters, example 145
- troubleshooting 147

keyword-driven tests

- converting from manual tests 136
- creating 136
- cross-browser testing 143
- versions 180, 206

keywords

- about 136
- adding 137
- assigning to configurations 258
- assigning to execution plans 272
- assigning to execution servers 479
- combining 137
- concepts 270
- creating 271
- deleting 137
- dynamic hardware provisioning 270
- execution servers 476
- implementing 137
- managing 137
- nesting 137
- opening 137
- parameters 137
- parameters, example 145
- removing from configurations 259
- removing from execution plans 272
- replacing 137
- sequences 137
- usages 195
- visibility 196

L

last executions

- deleting 327

LDAP

- importing groups 471
- synchronizing groups 471

libraries

- building 143, 200
- change track 208
- concept 195
- display in projects 196
- history 208
- introduction 195
- project access 196
- properties 195
- step properties 202
- uploading 144, 195
- visibility 196

licensing

- Silk Central 14

limitations

- VMware vCloud 483

limits

- data sources 113, 586

links

- attaching to test elements 189

list

- all issues 344

list of values

- configuration page 511

lists

- project baselines 526
- projects 526
- locating
 - assigned requirements 209
- locations
 - adding 474
 - deleting 475
 - editing 474
 - managing 474
- log files
 - level of detail 495
 - managing 495
 - servers 494
- logging in and out 24

M

- Mac
 - information service, installing 486
- maintaining
 - value lists 509
- managing
 - builds 90, 500
 - components 88, 502
 - device providers 484
 - keywords 137
 - locations 474
 - platforms 92, 503
 - products 85, 505
 - products and platforms 85, 500
 - project-wide notifications 396
 - projects 515
 - report subscriptions 424
 - report templates 487
 - requirements 61
 - tests 129
 - versions 89, 507
- managing requirements
 - quick start task 26
- managing tests
 - quick start task 28
- manual execution planning
 - assigning tests to manual testers 228
 - matching tests 229
 - overview 226
 - specify configurations 243
 - test assignment 236
 - test selection 228
- manual execution planning - walkthrough
 - adding manual testers 227
 - creating testing cycles 227
 - selecting tests and scheduling 228
 - user interface 226
 - walkthrough 226
- manual executions
 - planning 226
- manual test
 - blocked 253
 - configuring 129
 - creating 129
 - editing 129
- manual test steps

- adding data source values 191
- custom properties page 109, 540
- editing 129, 203
- properties page 109, 540
- manual testers
 - adding to configurations 259
 - assigning manual tests to 233, 239
 - moving tests to other manual testers 233, 240
- manual testing
 - attaching issues 251
 - attaching step result files 251
 - offline manual testing
 - overview 255
 - overview 244
- manual testing window
 - capturing screen images 249
 - code analysis 252
 - overview 244
 - overview area 244
 - recording videos 250
 - statuses of tests and test steps 253
 - test steps area 247
- manual tests
 - adding calls to shared steps 201
 - analyze results 319
 - assigning to manual testers 233, 239
 - business components, generating 130, 151
 - calls 201
 - converting to automated tests 130
 - converting to keyword-driven tests 136
 - importing, Excel 131
 - overview 129
 - step properties 202
 - versions 180, 206
- mapping
 - parameters, business process testing 153
 - requirements from Excel 79
- mapping issue states
 - issue tracking profiles 557
- mappings
 - VMware vCloud 483
- method coverage comparison
 - reports 454
- milestones
 - adding to testing cycles 232, 239
- mobile device
 - user interface 248
- mobile devices
 - assigning to configurations 260
 - selecting 274
 - selecting for manual test 252
- moving columns
 - inboxes 364
- moving tests
 - from one manual tester to another 233, 240
 - from one testing cycle to another 233, 240
- MSTest
 - test type 158, 637
- MSTest test
 - configuring 158, 637
 - creating 158, 637
 - editing 158, 637

- multi-edit
 - test attributes in grid view 222
 - test properties in grid view 222
- multiple TestPartner tests
 - adding 167, 645

N

- nesting
 - keyword 139, 196
 - keywords 137
- new execution server
 - dialog box 480
- new location
 - dialog box 475
- no schedules
 - specifying 292
- notations
 - parameters 188
- notification rules
 - defining 351
- notifications
 - entering parameters in subject 397
 - page 61, 122, 542
- NUnit
 - test type 159, 637
- NUnit test
 - configuring 159, 637
 - creating 159, 637
 - editing 159, 637

O

- obsolete
 - requirements 63
- obsolete requirements
 - convert to active 63
 - deleting 63
- Office import
 - business process tests, importing from Excel 155
 - editing the test mapping file 177
 - exporting shared steps objects 203
 - exporting tests 175
 - importing tests 176
 - manual tests, importing from Excel 131
 - test file, Excel 176
 - updating shared steps objects 204
 - updating tests 176
- offline manual testing
 - downloading tests 256
 - uploading results 256
- opening
 - keyword 139, 196
 - keywords 137
 - Silk Performer projects 312
- overall page-time trend
 - reports 455
- overall transaction busy-time trend
 - reports 456
- overriding
 - test parameters 266
- overview

- execution planning 16
- manual tests 129
- product 14
- requirements 15
- tests 15

P

- packages/namespaces
 - viewing code-coverage information 284, 445
- panels
 - adding to dashboard 20
- parallel execution
 - enabling 272
- parameters
 - adding to configurations 257
 - adding to execution plans 265
 - adding to tests 187
 - calling from JUnit tests 157, 636
 - clearing 187
 - creating custom 186
 - editing 187
 - execution plan run properties 185, 264
 - mapping, business process testing 153
 - notations 188
 - overriding test parameter 266
 - parameters within parameter values 188
 - predefined 185, 264
 - removing from configurations 258
 - removing from execution plans 266
 - replacing upon execution 188
 - specifying, business components 152
 - specifying, flows 153
 - tests 185
- pasting
 - test elements 214
- performance trend
 - reports 454
- performance trend reports
 - average page-time 454
 - average transaction busy-time 455
 - custom measure 456
 - overall page-time 455
 - overall transaction busy-time 456
- permissions
 - administration 465
 - executions 462
 - issues 463
 - libraries 461, 463
 - log files 465
 - manual execution planning 462
 - project settings 464
 - projects 464
 - quality goals 464
 - reports 464
 - requirements 461
 - tests 462
- plan files (.pln)
 - configuring 164, 642
 - running 164, 642
- platform configuration
 - page 93, 504

- platforms
 - activating 92, 504
 - adding 92, 503
 - deactivating 92, 504
 - deleting 93, 504
 - editing 92, 503
 - managing 92, 503
 - sorting list 92, 504
- predefined
 - parameters 185, 264
- printing
 - assigned tests 306
 - charts 426
 - tests 178
- printing manual tests 254
- priority
 - Atlassian Jira Software 45, 547, 609
- ProcessExecutor
 - test type 160, 638
- ProcessExecutor test
 - configuring 160, 638
 - creating 160, 638
 - editing 160, 638
- product
 - overview 14
- product settings
 - defining 366
- products
 - activating 86, 90, 505, 508
 - adding 85, 505
 - configuration page 86, 506
 - deactivating 86, 90, 505, 508
 - deleting 86, 506
 - editing 86, 505
 - managing 85, 505
 - sorting list 86, 505
- project baselines
 - activating 518
 - copying 518
 - creating 516
 - deactivating 518
 - deleting 519
 - editing 517
 - lists 526
 - overview 516
- Project Manager
 - user roles 458
- project settings
 - page 522, 591
 - requirement-related 36
 - test-related 85
- project-wide notifications
 - adding 396
 - deleting 397
 - displaying 398
 - editing 396
 - managing 396
- projects
 - accessing recent 527
 - activating 518
 - adding 515
 - adding quality goals 337
 - build information 524
 - build information files 524
 - comparing with baselines 525
 - configuring settings 527
 - copying 518
 - deactivating 518
 - defining inbox settings 367
 - deleting 519
 - deleting quality goals 337
 - editing 517
 - exporting 519
 - importing 520
 - lists 526
 - managing 515
 - overview 515
 - overview report 435
 - quality goals 336
 - quality goals page 336
 - see libraries 196
 - selecting 526
 - setting active 526
 - settings 527
 - tags 526
 - templates 591
 - working with (Issue Manager) 355
- properties
 - libraries 195
 - requirement 36, 534
- properties page
 - requirements 69
- proposed changes 348

Q

- quality goals
 - risk-based testing 337
- quick start tasks
 - overview 25

R

- Rational ClearQuest
 - adding issue tracking profiles 562, 596
 - editing issue tracking profiles 562, 596
 - issue tracking profiles 561, 595
- re-run
 - not executed tests 296
- reason
 - define status reason 591
 - status reason 590
- recent changes
 - filters 59, 121, 301, 531
- recent projects
 - accessing 527
 - switching 527
- recommendations
 - algorithm 147
- recommended keywords
 - keyword-driven test editor 147
- recording videos
 - manual testing window 250
- removing

- assigned requirements 209
- assigned tests from requirements 73
- charts 426
- data-driven properties 192
- execution plan parameters 266
- keywords from configurations 259
- keywords from execution plans 272
- parameters from configurations 258
- report templates 424
- requirements integration 56, 557, 619
- testers from configurations 259
- testers from execution plans 269
- removing columns
 - inboxes 364
- removing tests
 - from manual testers 234, 241
 - from testing cycles 234, 240
- reordering
 - columns in grid view 223
 - test elements 216
- replacing
 - keyword 139, 196
 - keywords 137
 - requirement properties 68
 - test properties 218
- report parameters
 - editing 353, 433
 - overview 433
- report properties
 - editing 432
 - editing (Issue Manager) 353
 - overview 432
- report templates
 - deleting 491
 - downloading 423, 490
 - editing 490
 - establishing database access 488
 - managing 487
 - overview 423
 - page 492
 - removing 424
 - selecting 423
 - updating sources 491
 - uploading 423, 490
 - Word 427
 - Word sample 428
- Reporter
 - user roles 458
- reporting
 - tables and views 406
- reports
 - accessing MRU 431
 - adding subreports 422
 - all related issues 65, 450
 - average page-time trend 454
 - average transaction busy-time trend 455
 - bookmarking 431
 - chart page 427
 - code coverage 453
 - code coverage trend 454
 - code-change impact 453
 - context-sensitive 445
 - creating 419
 - creating (Issue Manager) 352
 - creating and customizing 418
 - custom measure trend 456
 - data mart 398
 - data mart architecture 399
 - data mart overview 398
 - data page 434
 - default reports 445
 - deleting subreports 422
 - displaying charts 426
 - downloading templates 423
 - editing parameters 433
 - editing properties 432
 - embedding 424
 - execution plan run comparison 452
 - execution plan run errors 453
 - execution planning 452
 - execution status overview 453
 - generating 430
 - generating (Issue Manager) 352
 - generating code-change impact 283, 444
 - global 418
 - how to create 399
 - issues per component 453
 - managing subscriptions 424
 - manual tests 211, 451
 - method coverage comparison 454
 - overall page-time trend 455
 - overall transaction busy-time trend 456
 - overview 398
 - parameters page 434
 - performance trend 454
 - printing charts 426
 - project overview 335, 435, 457
 - properties page 432
 - remote access 424
 - removing charts 426
 - removing templates 424
 - report page 434
 - reporting area 418
 - requirement documents 65, 450
 - requirement progress 65, 449
 - requirement status 64, 449
 - requirements 64, 449
 - saving 430
 - selecting templates 423
 - subscribing to a report 424
 - subscriptions 424
 - test baseline comparison 212, 452
 - test progress 211, 450
 - test run comparison 212, 452
 - test status 210, 451
 - tests 210, 450
 - tests not passed per execution plan 453
 - toolbar functions 431
 - uploading templates 423
 - viewing 430
 - viewing as PDF 431
 - writing advanced SQL queries 421
- repository configuration
 - initializing (Issue Manager) 356

- Request entity is too large 520
- requirement
 - properties 36, 534
- requirement attachments
 - deleting 71
 - overview 70
 - viewing 71
- requirement properties
 - custom 39, 537
 - delete 41, 539
 - editing 41, 539
 - finding 69
- requirement property types 39, 110, 532, 538
- requirement reports
 - all related issues 65, 450
 - documents 65, 450
 - progress 65, 449
 - status 64, 449
- requirements
 - activate obsolete 63
 - all related issues report 65, 450
 - assigned tests page 74
 - assigning tests from grid view 72
 - associating tests 72
 - attaching files 70
 - attaching links 71
 - attachments 70
 - attachments page 72
 - calculated properties 37, 535
 - changes 83
 - clearing flags 67
 - collapsing tree 81
 - coverage 65
 - coverage modes 65
 - coverage page 65
 - creating 62
 - creating a calculated property 37, 536
 - creating custom properties 39, 537
 - custom properties 39, 537
 - custom properties page 36, 534
 - delete properties 41, 539
 - deleting attachments 71
 - deleting obsolete 63
 - document reports 65, 450
 - document view 83
 - downloading attachments 71
 - editing 62
 - editing external properties 54, 555, 617
 - email notifications for automatic synchronization 43, 545, 607
 - Excel file 79
 - expanding tree 81
 - finding properties 69
 - flags 66
 - generating tests 74
 - history 83
 - history page 84
 - internal property values 38, 537
 - locating assigned tests 73
 - managing 61
 - manually assigning tests 73
 - marking as obsolete 63
 - overview 15, 36
 - progress reports 65, 449
 - properties 68
 - properties page 69
 - removing assigned tests 73
 - replacing custom properties 68
 - reports 64, 449
 - setting flags 67
 - status reports 64, 449
 - synchronizing 41, 543, 605
 - synchronizing across tools 43, 544, 606
 - synchronizing based on schedules 43, 544, 607
 - test coverage status 65
 - tooggling coverage modes 65
 - toolbar functions 82
 - tracking history 84
 - tree 80
 - updating from Excel 79
 - updating from Word 77
 - viewing attachments 71
 - viewing external properties 54, 555, 618
 - viewing recent changes 83
 - Word file 77
- requirements coverage
 - modes 65
- requirements import
 - overview 76
- requirements integration
 - deleting property mapping 55, 556, 619
 - disabling 55, 557, 619
 - editing property mapping 55, 556, 618
 - removing 56, 557, 619
- requirements management tools
 - overview 41, 543, 605
- requirements tree
 - collapsing 81
 - expanding 81
 - filtering 81
- rerun
 - not executed tests 296
- resetting
 - grid view 225
- resizing
 - columns in grid view 223
- REST API
 - using, external executions 298
- restoring default settings
 - grid view 225
- result files
 - deleting 303
 - WSH 169, 647
- results
 - deleting 303
 - external, managing 297
 - screenshots 302
 - videos 302
- reuse
 - test steps 202
- reusing
 - test packages 128
 - tests 172, 214
- reverting

- test packages to tests 128
- risk based testing
 - about quality goals 336
 - adding quality goals 337
 - deleting quality goals 337
 - quality goals reference page 336
- risk-based testing
 - overview 335
 - quality goals 337
- Role Assignment Manager
 - user roles 458
- roles settings
 - page 460
- routing rules
 - adding 370
 - deleting 371
 - editing 370
 - overriding automatic preferences 372
 - reordering 371
 - setting up 368
 - specifying 369
- run
 - dialog box 296
- run comparison 319
- running
 - plan files (.pln) 164, 642
- runs
 - page 324

S

- sample database
 - exploring 342
- SAP Solution Manager
 - configuring 53, 554, 565, 599, 616
 - issue tracking profiles 53, 554, 565, 599, 616
 - system requirements 52, 554, 565, 599, 616
- SAP Solution Manager integration 52, 553, 565, 599, 616
- saving
 - reports 430
- Scenario
 - generating tests 50, 551, 613
- schedules
 - activating 513
 - adding 512
 - adding definite runs 294
 - adding exclusions 293
 - configuration page 514
 - configuring 511
 - creating custom 292
 - deactivating 513
 - definite runs 291, 294
 - deleting 513
 - deleting definite runs 294
 - deleting exclusions 293
 - editing 512
 - editing definite runs 294
 - exclusions 291, 293
 - scheduling definite runs 514
 - scheduling exclusions 513
 - specifying global 292
 - specifying no schedules 292

- scheduling
 - definite runs 514
 - execution plans 291
 - schedule exclusions 513
- screen capturing
 - enabling for tests 302
- search
 - global 18
 - in Silk Central 18
- selecting
 - mobile devices 274
 - mobile devices for manual test 252
 - report templates 423
- Selenium
 - keyword libraries 143, 200
 - test type 156, 634
- server log files
 - analyzing 494
 - changing level of detail 495
 - deleting 495
 - downloading 494
 - level of detail 495
 - managing 495
- servers
 - log files 494
- setting flags
 - requirements 67
- settings
 - attribute types 39, 110, 532, 538
 - attributes 109, 531
 - attributes page 112, 534
 - browser 24
 - change notification 60, 121, 541
 - creating custom requirement properties 39, 537
 - filters 56, 117, 298, 527
 - projects 527
 - requirement property types 39, 110, 532, 538
 - VMware vCloud 483
- shared
 - tests 172
- shared objects
 - adding calls 201
 - calls 201
 - overview 201
 - steps 132, 205
 - test steps 132, 205
 - versions 180, 206
 - visibility 196
- shared steps
 - creating 202
 - extracting 202
- shared steps objects
 - exporting to Excel 203
 - import and update 203
 - updating from Excel 204
 - usages 195
- shared tests
 - updating 173
 - versions 173
- sharing
 - tests 172, 214
 - tests, consuming project 174, 215

- show
 - issues 344
- Silk Central
 - concepts 14
 - installing 14
 - licensing 14
 - logging in 24
 - logging out 24
 - login 24
 - logout 24
 - welcome 9
 - workflow 14
- Silk Central Administration Module
 - relationship with Issue Manager 356
- Silk Performer
 - accessing parameters 313
 - analyzing test results 314
 - downloading test result packages 315
 - editing test properties 314
 - executing attended tests 313
 - Performance Explorer 316
 - projects 311
 - test type 162, 640
 - uploading test results 315
- Silk Performer .NET Explorer
 - test type 161, 639
- Silk Performer .NET Explorer test
 - configuring 161, 639
- Silk Performer projects
 - downloading 312
 - opening 312
- Silk Performer test
 - configuring 162, 640
 - creating 162, 640
 - editing 162, 640
- Silk Test
 - exporting results from 348
- Silk Test Classic
 - adding AUT host 284, 316
 - agent under test 317
 - automated execution of data-driven test cases 317
 - executing automated tests 317
 - logs 317
 - test type 162, 640
 - time-out settings 317
- Silk Test Classic plan
 - configuring 164, 642
- Silk Test Classic plan tests
 - creating 164, 642
 - editing 164, 642
- Silk Test Classic tests
 - configuring 162, 641
 - creating 162, 641
 - editing 162, 641
- Silk Test Intelligent Assistant 347
- Silk Test results
 - importing to Issue Manager 347
- Silk Test Workbench
 - adding source control profiles 101, 578, 626
 - editing source control profiles 101, 579, 627
 - source control profiles 100, 578, 626
 - test parameters 166, 189, 644
 - test type 164, 643
- Silk Test Workbench test
 - configuring 165, 643
 - creating 165, 643
 - editing 165, 643
- Silk4J
 - test type 156, 634
- Silk4NET
 - test type 159, 637
- Simulation
 - generating tests 50, 551, 613
- SOAP API
 - Issue Manager 354
- software quality optimization 14
- sorting
 - builds 91, 501
 - components 89, 503
 - platforms 92, 504
 - products 86, 505
 - test folders 126
 - tests in grid view 223
 - values in value lists 510
 - versions 90, 508
- source control
 - page 108, 586
- source control profiles
 - AccuRev 94, 571, 619
 - deleting 108, 585
 - Git 97, 575, 623
 - overview 93, 571
 - page 108, 586
 - Silk Test Workbench 100, 578, 626
 - StarTeam 102, 579, 627
 - Subversion 103, 581, 629
 - SVN 103, 581, 629
 - Team Foundation Server 99, 576, 624
 - TestPartner 105, 582, 631
 - TFS 99, 576, 624
 - TFS, adding 99, 576, 625
 - UNC 106, 584, 632
 - Universal Naming Convention 106, 584, 632
 - VFS 95, 573, 621
 - Virtual File System 95, 573, 621
- specifying
 - global schedules 292
 - no schedules 292
- SSL
 - handling self-signed certificates 651
- standard issue fields
 - customizing 374
- StarTeam
 - adding issue tracking profiles 569, 603
 - adding source control profiles 102, 579, 627
 - editing issue tracking profiles 569, 604
 - editing source control profiles 102, 580, 628
 - issue tracking profiles 568, 602
 - source control profiles 102, 579, 627
- starting
 - execution plan 296
- state properties 385
- status
 - define reason 591

- reason 590
- status calculation
 - execution plans 263
- status reason
 - concepts 590
 - define 591
- step properties
 - creating 109, 540
 - custom 108, 540
 - deleting 109, 540
 - editing 109, 540
 - libraries 202
 - manual tests 202
 - page 109, 540
- structuring
 - tests 123
- STW.EXE
 - prerequisites, Silk Central 164, 643
- STW.EXE, prerequisites 164, 643
- subreports
 - adding 353
 - adding to reports 422
 - deleting from reports 422
 - overview 422
- subscribing
 - to a report 424
- subscriptions 424
- subversion
 - adding source control profiles 103, 581, 629
- Subversion
 - editing source control profiles 104, 582, 630
 - source control profiles 103, 581, 629
- success conditions
 - editing 194
 - tests 193
- SuperUser
 - user roles 458
- supported
 - test types 129
- SVN
 - adding source control profiles 103, 581, 629
 - editing source control profiles 104, 582, 630
 - source control profiles 103, 581, 629
- switching
 - recent projects 527
- switching modes
 - test coverage 65
- synchronize
 - virtual execution servers 478
- Synchronize Current Runs
 - dialog 310
- synchronizing
 - current run 310
 - data sources 116, 589
 - groups from LDAP 471
 - internal and external issue states 341
 - requirements 41, 543, 605
 - requirements across tools 43, 544, 606
- synchronizing requirements
 - based on schedules 43, 544, 607
- system requirements
 - SAP Solution Manager 52, 554, 565, 599, 616

T

- tab labels
 - customizing 374
- tables
 - data mart 406
 - reporting 406
- tagging
 - builds 91, 501
- tags
 - project-level 526
- task management tools
 - integrating 499
- Team Foundation Server
 - adding issue tracking profiles 567, 601
 - editing issue tracking profiles 568, 602
 - installing proxy service 566, 600
 - issue tracking profiles 566, 600
 - source control profiles 99, 576, 624
- test assets
 - workflow 133
- test assignment
 - overview 236
- test attachments
 - deleting 190
 - viewing 190
- test automation
 - overview 136
- test containers
 - adding 123
 - adding links 124
 - editing 125
 - filtering 216
 - overview 123
- test coverage
 - modes 65
 - switching modes 65
- test coverage page
 - requirements 65
- test coverage status
 - requirements 65
- test elements
 - copy to other projects 214
 - copying 214
 - copying between projects 214
 - cutting 214
 - deleting 214
 - pasting 214
 - reordering 216
- test export
 - Excel 174
- test folders
 - adding 125
 - editing 125
 - filtering 216
 - overview 125
 - sorting 126
- test import
 - Excel 174
- Test Manager
 - user roles 458
- test mapping file

- editing 177
- test packages
 - cleaning up 129
 - copying 128
 - creating 126
 - External IDs 126
 - overview 126
 - reusing 128
 - reverting to test 128
- test parameters
 - Silk Test Workbench 166, 189, 644
- test properties
 - configuring 182
 - finding 217
 - replacing 218
- test reports
 - baseline comparison 212, 452
 - manual tests 211, 451
 - progress 211, 450
 - run comparison 212, 452
 - status 210, 451
- test run results
 - compare 319
- test runs
 - analyzing 319
 - deleting results 320
 - details 322
 - viewing details 320
- test selection
 - overview 228
- test status
 - changing 320
- test steps
 - calling 201
 - downloading attachments 132, 205
- test type
 - .NET Explorer 161, 639
 - JUnit 156, 634
 - MSTest 158, 637
 - NUnit 159, 637
 - ProcessExecutor 160, 638
 - Selenium 156, 634
 - Silk Performer 162, 640
 - Silk Performer .NET Explorer 161, 639
 - Silk Test Classic 162, 640
 - Silk Test Workbench 164, 643
 - Silk4J 156, 634
 - Silk4NET 159, 637
 - TestPartner 166, 645
 - UFT Developer 156, 634
- test types
 - supported 129
- testbook 254
- Tester
 - user roles 458
- testing Apple Safari
 - information service, installing 486
- testing cycle capacity 231, 238
- testing cycles
 - adding milestones 232, 239
 - assigning tests 232, 239
 - creating 230, 237
 - deleting 236, 243
 - finishing 236, 243
 - hiding 235, 242
 - importing tests
 - from execution plans to testing cycles 234, 241
 - importing tests from execution plans to 234, 241
 - moving tests to other testing cycle 233, 240
 - properties page 306
 - removing tests 234, 240
 - removing tests from manual testers 234, 241
 - starting 235, 242
 - testing cycles
 - showing 235, 242
 - version and build 235, 242
- testing cycles properties page 306
- TestPartner
 - adding source control profiles 105, 583, 631
 - editing source control profiles 106, 583, 631
 - source control profiles 105, 582, 631
 - test type 166, 645
- TestPartner tests
 - configuring 166, 645
 - creating 166, 645
 - editing 166, 645
- tests
 - adding parameters 187
 - assigning attributes 184
 - assigning existing issues in grid view 225
 - assigning from grid view to execution plans 287
 - assigning requirements 208
 - assigning to execution plans through filter 288
 - assigning to requirements from grid view 72
 - attaching files 189
 - attaching links 189
 - attachments 189
 - attributes 184
 - baseline comparison report 212, 452
 - changing run status 320
 - cleanup 266
 - contents 218
 - deleting attachments 190
 - deleting attributes 184
 - document view 221
 - downloading attachments 190
 - editing 174
 - editing attributes 184
 - enabling screen capturing 302
 - enabling video capturing 303
 - executing 295
 - executing trial runs 194
 - exporting to Excel 175
 - filtering in grid view 223
 - generating from requirements 74
 - grid view 221
 - grouping in grid view 223
 - history page 182
 - importing from Excel 176
 - integration default nodes 216
 - issues page 179
 - managing 129
 - manual test reports 211, 451

- manually assigning to execution plans 288
- manually assigning to requirements 73
- multi-edit attributes 222
- multi-edit properties 222
- overview 15, 85
- parameters 185
- printing 178
- progress reports 211, 450
- properties page 182
- removing assigned requirements 209
- reports 210, 450
- run comparison reports 212, 452
- run details 322
- runs page 178
- screen captures 302
- setup 266
- shared 172
- status reports 210, 451
- steps 132, 205
- structuring 123
- success conditions 193
- test containers 123
- test steps 132, 205
- toolbar functions 219
- tracking history 182
- tree 213
- updating from Excel 176
- versions 180, 206
- video captures 302
- viewing assigned executions 212
- viewing attachments 190
- viewing recent changes 181
- tests tree
 - collapsing 213
 - expanding 213
- text export
 - rules 175
- text import
 - rules 175
- text update
 - rules 175
- TFS
 - adding issue tracking profiles 567, 601
 - adding source control profiles 99, 576, 625
 - editing issue tracking profiles 568, 602
 - editing source control profiles 100, 577, 625
 - installing proxy service 566, 600
 - issue tracking profiles 566, 600
 - source control profiles 99, 576, 624
- time zones
 - overview 514
- toolbar
 - execution planning 307
 - requirements 82
 - tests 219
- toolbar functions
 - reports 431
- tracking
 - overview 327
- tracking history
 - requirements 84
 - tests 182

- trial runs
 - tests 194
- troubleshooting
 - data mart 405
 - keyword-driven testing 147

U

- UFT 155, 634
- UFT tests
 - configuring 155, 634
 - creating 155, 634
 - editing 155, 634
- UNC
 - adding source control profiles 106, 584, 632
 - editing source control profiles 107, 585, 633
 - source control profiles 106, 584, 632
- Unified Functional Testing 155, 634
- Universal Naming Convention
 - source control profiles 106, 584, 632
- update
 - virtual execution servers 478
- updating
 - build informations 524
 - report template sources 491
 - shared steps objects from Excel 204
 - shared tests 173
 - tests from Excel 176
- updating from Excel
 - requirements 79
- updating from Word
 - requirements 77
- updating tests
 - Excel 174
- uploading
 - keyword libraries 144, 195
 - libraries 144, 195
 - report templates 423, 490
 - Silk Performer test results 315
- uploading Excel files
 - data sources 116, 589
- user account properties
 - initial issue state 365
 - issue related 365
- user accounts
 - adding 467
 - assigning groups 467
 - assigning roles 467
 - deleting 468
 - editing 366, 468
 - maintaining 467
 - security settings 365
 - setting up for issue management 364
- user accounts and groups
 - overview 467
- user groups
 - setting up for Issue Manager 359
- user interface
 - mobile device 248
 - overview 16
 - tour 16
 - using 16

- user roles
 - adding 459
 - copying 460
 - deleting 460
 - description 458
 - editing 459
 - overview 458
 - permissions 461
- user roles and permissions
 - overview 458
- user security settings
 - Issue Manager 365
- user settings
 - page 469
- using
 - user interface 16
- using Issue Manager
 - assessing the interface 357
 - assessing the workflows 357
 - gathering information about people 357
 - gathering information about products 357

V

- value lists
 - activating 510
 - adding 509
 - deactivating 510
 - deleting 510
 - editing 509
 - maintaining 509
 - overview 509
 - sorting values 510
- values
 - activating 510
 - deactivating 510
 - deleting 511
 - editing 509
- vCloud
 - adding profile 481
 - deleting profile 483
 - editing profile 482
 - limitations 483
 - mappings 483
 - settings 483
- version
 - specifying for testing cycle 235, 242
- version and build
 - editing 253
- versioning
 - tests during Excel import 177
 - tests during Excel update 177
- VersionOne
 - integrating 499
- versions
 - adding 89, 507
 - comparing 181, 207
 - creating 180, 207
 - deleting 90, 508
 - differences 181, 207
 - differing 181, 207
 - editing 89, 508

- managing 89, 507
- reverting 181, 207
- shared tests 173
- sorting list 90, 508
- viewing 180, 207

VFS

- adding source control profiles 96, 573, 621
- editing source control profiles 96, 574, 622
- source control profiles 95, 573, 621

video capturing

- enabling for tests 303

viewing

- assigned executions 212
- audit log 493
- external properties 54, 555, 618
- requirement attachments 71
- test attachments 190
- test run details 320
- viewing execution result files
 - visual tests 165, 644
- viewing recent changes
 - requirements 83
 - tests 181

views

- reporting 406

Virtual File System

- source control profiles 95, 573, 621

visual tests

- viewing execution result files 165, 644

VMware vCloud

- adding profile 481
- deleting profile 483
- editing profile 482
- limitations 483
- mappings 483
- settings 483

W

Windows Script Host

- log information 169, 647
- parameters 169, 647
- properties 168, 647
- result files 169, 647
- supported script languages 168, 646
- viewing returned information 169, 647

Windows Script Host tests

- overview 167, 646

Word

- importing requirements 76

Word file

- requirements 77

workflow

- action properties 386
- add state owners and permissions 384
- bug 379
- current state 386
- customize actions, reason codes, action dialogs 385
- customizing states 385
- developing 381
- documentation issue 380
- drawing a state diagram 382

- enhancement 380
- name actions with verbs 383
- new state list 387
- optimizing 383
- overview 14
- permissions tab 390
- prepare data entry sheet 384
- standard action fields tab 388
- state customization 384
- state properties 385
- user-defined action fields tab 389

working with issues

- actions for 349

WSH

- overview 167, 646
- parameters 169, 647
- properties 168, 647
- supported script languages 168, 646
- viewing returned information 169, 647

WSH test

- configuring 167, 646
- creating 167, 646
- editing 167, 646