Introduction to ALM, UFT, VuGen, and LoadRunner

This course introduces students to the Application Lifecycle Management line products
Introduction to ALM, UFT, VuGen, and LoadRunner

The objective is to share the know-how to use the Micro Focus® UFT 12.00 application as an automated functional testing tool. You can use the point and click interface to record and play back tests, add synchronization points and verification steps, and create multiple action tests. ALM 12.00, on the other hand, educates students to manage quality information throughout the development cycle—from constructing requirements, designing and executing tests, through monitoring defects. LR 12.00 is the tool used to test the performance of the application. The course covers topics about the Virtual User Generator (VuGen), Controller, and Analysis tools. You learn to work with the graphs to display data after a test is executed.

Unified Functional Testing (UFT) 12.00, Application Lifecycle Management (ALM) 12.00, LoadRunner (LR) 12.00, and Virtual User Generator (VuGen) 12.00.

This course is designed for:

■ Manual Testers who want to move to automated testing tools
■ College students

Course Objectives
At the end of the course, you should be able to:

■ Create basic scripts from a manual test case
■ Parameterize tests to run with multiple sets of data
■ Create and reuse modular actions
■ Use the Object Repository Manager
■ Resolve object recognition problems

Use Local System Monitoring
■ Retrieve and use the properties of an object
■ Create programmatic descriptions
■ Retrieve data from application objects
■ Create scripts that access data from external sources
■ Create new subroutines and functions
■ Use the Function Library editor

Prerequisites/Recommended Skills
To be successful in this course, you should have a working knowledge of:

■ Windows
■ Websites and browsers
■ Programming and scripting languages
# Course Topics

## Modules | Objectives

### Unified Functional Testing

#### Module 1—Course Overview
- Identify the contents and objectives of the course
- Define the class schedules and class logistics
- Identify the related courses
- Discuss the lab environment details

#### Module 2—Software Overview
- Describe the advantages of UFT as a testing tool
- Preview the UFT Start Page and Help menus
- Recognize the sample applications used in the labs
- Identify resources for getting assistance

#### Module 3—Preparing to Record
- Identify functional testing principles and the benefits of automated testing
- Navigate the typical GUI testing workflow
- Document the steps of a business process
- Prioritize business processes using effective criteria
- Gather sufficient test data
- Prepare the test environment for automated testing

#### Module 4—Creating a Basic Test
- Create a basic test from a manual test case
- Run a test and check for errors
- Save a test
- View test results

#### Module 5—Working with Objects
- Identify objects
- Define a UFT for GUI Testing object
- Identify objects in UFT for GUI Testing
- Use the object repository to manage objects in UFT for GUI Testing

#### Module 6—Adding Synchronization
- Define synchronization in UFT for GUI Testing
- Identify the uses of synchronization in UFT for GUI Testing
- Add a synchronization step for a specified object

#### Module 7—Verifying with Standard Checkpoints
- Define standard checkpoints
- Add standard checkpoints to a test
- Use a regular expression to add flexibility to a standard checkpoint

#### Module 8—Using Parameters
- Identify and use different parameter types
- Insert an input parameter
- Insert an output parameter
- Parameterize a checkpoint
- Evaluate test results for iterative tests
- Automatically parameterize steps

#### Module 9—Building Multiple, Reusable Actions
- Identify actions in GUI testing
- Identify action types
- Identify action and test iterations
- Identify calls to existing actions and copies of actions
- Share values using the global data table
- Call actions with parameters
- Store action return values
- Create multiple actions from a single action
- Create a new action
- Call a reusable action from another test
- Use local and global data sheets
- Resolve missing actions

*Continued on the next page*
<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
</table>
| **Module 10—Adding Steps without Recording** | - List the types of steps that can be added to a test without using the record feature  
- Use conditional statements in a test  
- Use the Step Generator  
- Use the reporter object to report events in the test results |
| **Module 11—Creating Tests on a Web Application** | - Record and run a test on a web application  
- Insert a text checkpoint in a test for a web application |
| **Module 12—Writing Custom Checkpoints** | - Create a custom checkpoint  
- Compare captured parameter values with expected values  
- Use debug tools to investigate the causes of failure  
- Verify that a reported error message reflects the state of the test |
| **Module 13—Utilizing a Shared Object Repository** | - Identify the types of object repositories  
- Manage shared object repositories using the Object Repository Manager  
- Use visual relation identifiers |
| **Appendix 1—Using Recovery Scenarios** | - Identify exceptions in a test  
- Create a recovery scenario  
- Associate a recovery scenario with a test  
- Set an optional step in a test |
| **Appendix 2—Object Identification Techniques** | - Configure object identification  
- Describe mandatory and assistive properties  
- Use ordinal identifiers  
- Use smart identifiers  
- Describe when to use Smart Identification  
- Use the Smart Identification process  
- Describe how UFT for GUI Testing uses Smart Identification  
- Use Case Scenario Test object mapping for unidentified or custom classes |

---

**Unified Functional Test Advanced**

<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
</table>
| **Module 1—Introduction to the Editor** | - Identify the advantages of the Editor  
- Translate steps between the Keyword view and the Editor view  
- Explain how VBScript and objects are relevant to UFT for GUI Testing  
- List common test objects and methods used in UFT for GUI Testing |
| **Module 2—Using the Editor** | - Use the step generator to create new steps in your text  
- Enter steps manually in the Editor  
- Use statement completion (IntelliSense)  
- Trace and debug tests |
| **Module 3—Using VBScript** | - Create constants and variables to hold important values  
- Format steps and create comments for clarity  
- User operators to modify or compare values  
- Build conditional statements  
- Call built-in functions from the VBScript function library |
| **Module 4—Working with Object Properties** | - Retrieve any property of any object in the application during a test run  
- Retrieve and set properties in the object repository  
- Use additional properties of the Report utility object |
| **Module 5—Working with Dynamic Data** | - Retrieve useful properties from list-type objects  
- Describe the VBScript looping statements  
- Use additional methods to explore the Data Table |

*Continued on the next page*
<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
</table>
| Module 6—Working with Dynamic Objects | - Identify types of dynamic objects  
- Manage dynamic properties of objects in the object repository  
- Build a programmatic description for an object  
- Create and use a Description object  
- Retrieve a collection of child objects from a parent object |
| Module 7—Retrieving External Data | - Build scripts that access data from external sources  
- Import data from and export data to a Microsoft Excel worksheet  
- Use the Connection and RecordSet objects to query a database  
- Import and export data to text files |
| Module 8—Handling GUI Testing Exceptions | - Describe exception in a test  
- Handle positive and negative test data  
- Handle exceptions programmatically |
| Module 9—Creating New Procedures | - Identify the advantages of creating a procedure in a test  
- Create new subroutines and functions  
- Build a function library and associate it with a test  
- Register a procedure with an object class  
- Use a function as a recovery operation in a Recovery Scenario |

**Application Life Cycle Management**

<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
</table>
| Module 1—Course Overview | - Introduction  
- Course overview  
- Course objectives  
- Introduction to the case study  
- Introduction to the lab environment |
| Module 2—Introduction to Application Lifecycle Management | - Describe the Micro Focus Application Lifecycle Management (ALM) product  
- Describe the ALM hierarchy and roadmap  
- Navigate through the ALM modules  
- Describe the key features and benefits of ALM  
- Identify what’s new in ALM version 12.0  
- Identify additional helpful resources |
| Module 3—Working with ALM Clients | - Identify the ALM clients  
- Use the ALM 12 web client  
- Work with requirements  
- Work with defects  
- Perform web client customization  
- Design and manage forms  
- Work with business rules |
| Module 4—Working with Releases | - Identify the relationship between a line of business (LOB), applications, releases, and cycles  
- Create a release tree  
- Recognize the significance of assigning requirements to releases and cycles  
- Recognize the significance of assigning tests to releases and cycles |
| Module 5—Project Planning and Tracking | - Understand the terminology used in PPT  
- Work with PPT including:  
  - Define scope items  
  - Assign content to scope items  
  - Define and configuring milestones  
  - Assign and configure Key Performance Indicators (KPIs) |

Continued on the next page
<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 5—Project Planning and Tracking</td>
<td>- Create a custom KPI</td>
</tr>
<tr>
<td></td>
<td>- Calculate KPIs</td>
</tr>
<tr>
<td></td>
<td>- View the scorecard</td>
</tr>
<tr>
<td></td>
<td>- Use PPT best practices</td>
</tr>
<tr>
<td></td>
<td>- Troubleshoot PPT</td>
</tr>
<tr>
<td>Module 6—Working with Requirements and</td>
<td>- Specify requirements</td>
</tr>
<tr>
<td>Analyzing Risk</td>
<td>- Identify the characteristics of a useful requirement</td>
</tr>
<tr>
<td></td>
<td>- Add requirements to a project</td>
</tr>
<tr>
<td></td>
<td>- Create a requirements tree</td>
</tr>
<tr>
<td></td>
<td>- Assign requirements to releases and cycles</td>
</tr>
<tr>
<td></td>
<td>- Add traceability links using traceability</td>
</tr>
<tr>
<td></td>
<td>- Add traceability links between requirements</td>
</tr>
<tr>
<td></td>
<td>- Perform risk analysis for requirements</td>
</tr>
<tr>
<td>Module 7—Test Planning</td>
<td>- Organize subjects and tests in a Test Plan tree</td>
</tr>
<tr>
<td></td>
<td>- Create tests that define the steps for testing an application</td>
</tr>
<tr>
<td></td>
<td>- Use parameters in tests</td>
</tr>
<tr>
<td></td>
<td>- Generate test scripts from design steps</td>
</tr>
<tr>
<td></td>
<td>- Define test configurations</td>
</tr>
<tr>
<td></td>
<td>- Generate a live analysis graph from a Test Plan tree</td>
</tr>
<tr>
<td>Module 8—Requirements Coverage</td>
<td>- Create test coverage in the Requirements module</td>
</tr>
<tr>
<td></td>
<td>- Create requirement coverage in the Test Plan module</td>
</tr>
<tr>
<td></td>
<td>- Create requirement coverage using test configurations</td>
</tr>
<tr>
<td></td>
<td>- Analyze cycle progress</td>
</tr>
<tr>
<td></td>
<td>- Track cycle progress with the test set folders</td>
</tr>
<tr>
<td>Module 9—Test Execution</td>
<td>- Create and organize folders in a Test Sets tree</td>
</tr>
<tr>
<td></td>
<td>- View all test runs for a project</td>
</tr>
<tr>
<td></td>
<td>- Create test sets</td>
</tr>
<tr>
<td></td>
<td>- Add tests and test configurations into test sets</td>
</tr>
<tr>
<td></td>
<td>- Link test set folders to releases and cycles</td>
</tr>
<tr>
<td></td>
<td>- Manage test execution flow and test dependencies</td>
</tr>
<tr>
<td></td>
<td>- Execute manual and automated tests</td>
</tr>
<tr>
<td></td>
<td>- Record and review the results of test executions</td>
</tr>
<tr>
<td>Module 10—Lab Management</td>
<td>- Identify the motivation behind lab management</td>
</tr>
<tr>
<td></td>
<td>- Identify the concepts of lab management</td>
</tr>
<tr>
<td></td>
<td>- Manage lab resources</td>
</tr>
<tr>
<td></td>
<td>- Schedule and execute tests</td>
</tr>
<tr>
<td></td>
<td>- Understand Application Under Test (AUT) environments</td>
</tr>
<tr>
<td>Module 11—Build Verification</td>
<td>- Describe Build Verification functionality</td>
</tr>
<tr>
<td></td>
<td>- Create a Build Verification suite</td>
</tr>
<tr>
<td></td>
<td>- Add functional test sets to the suite</td>
</tr>
<tr>
<td></td>
<td>- Add performance tests to the suite</td>
</tr>
<tr>
<td></td>
<td>- Run the Build Verification suite</td>
</tr>
<tr>
<td></td>
<td>- View test results</td>
</tr>
<tr>
<td>Module 12—Micro Focus Sprinter 12.0</td>
<td>- Review and understand the manual test lifecycle</td>
</tr>
<tr>
<td></td>
<td>- Understand Sprinter features and functionality</td>
</tr>
<tr>
<td></td>
<td>- Use Sprinter</td>
</tr>
<tr>
<td></td>
<td>- Author tests</td>
</tr>
<tr>
<td></td>
<td>- Perform exploratory testing</td>
</tr>
<tr>
<td></td>
<td>- Define storyboarding</td>
</tr>
<tr>
<td></td>
<td>- Use Sprinter to log defects</td>
</tr>
</tbody>
</table>

Continued on the next page
<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
</table>
| Module 13—Defect Tracking     | Log defects  
  - Search and review defects  
  - Track defects throughout their lifecycle  
  - Associate defects with entities |
| Module 14—Version Control     | Describe version control functionality  
  - Check out entities  
  - Check in entities  
  - View version history  
  - Compare versions  
  - Promote an older version |
| Module 15—Library Management | Define a library  
  - Define a baseline  
  - Compare baselines  
  - Pin a test set to a baseline |
| Module 16—Asset Sharing (Libraries) | Define an asset library  
  - Import an asset library  
  - Work in parallel with source and target projects  
  - Compare imported assets from the source or target  
  - Synchronize assets  
  - Define cross-project sharing  
  - Share components  
  - Use the ALM Synchronizer  
  - Define a Hub project  
  - Create user-defined fields  
  - Explain the defect sharing workflow  
  - Apply best practices for reusing assets |
| Module 17—Exporting from Excel to ALM | Identify the types of data that you can export to Quality Center  
  - Install an add-in for Microsoft Excel to allow data to be exported to Quality Center  
  - Format requirements, test plan, and defects data in Microsoft Excel files  
  - Execute the Export wizard in Excel  
  - Verify the exported data in Quality Center |
| Module 18—Reporting and Analysis | Describe reporting and analysis in ALM  
  - Identify the features of the dashboard  
  - Create dashboard folders and pages  
  - Configure the dashboard  
  - View a dashboard page  
  - Generate reports and graphs  
  - Analyze reports and graphs  
  - Create and view project reports  
  - Generate formatted project documentation and Excel reports  
  - Share graphs that you can open without the ALM client |
| Module 19—Cross-Project Reporting | Describe reporting and analysis in ALM  
  - Identify the features of the dashboard  
  - Create dashboard folders and pages  
  - Configure the dashboard  
  - View a dashboard page  
  - Generate reports and graphs |

Continued on the next page
<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
</table>
| Module 19—Cross-Project Reporting continued continued | ■ Analyze reports and graphs  
■ Create and view project reports  
■ Generate formatted project documentation and Excel reports  
■ Share graphs that you can open without the ALM client |
| Module 20—Using ALM             | ■ Using ALM from A to Z                                                     |

<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Runner 12.x</td>
<td></td>
</tr>
</tbody>
</table>
| Module 1—Course Overview of LoadRunner 12.0 Essentials | ■ Identify the contents and objectives of the course  
■ Define the class schedule and class logistics  
■ Identify the related courses  
■ Discuss the lab environment details |
| Module 2—Introduction to Micro Focus LoadRunner (LR) 12.0 | ■ Explain the need for load testing  
■ Describe various types of performance test objectives  
■ Identify the steps of the LR methodology  
■ Define a scenario in the context of LR  
■ Identify strategies for creating effective scenarios |
| Module 3—What’s New in LR 12.0? | ■ Identify the key new features of LR 12.0, including:  
- Enhanced mobile testing  
- Continuous testing  
- Deploying Load Generators (LGs) in the cloud  
- Building a community  
- Communication interface enhancements  
- User experience and miscellaneous enhancements |
| Module 4—Planning an Effective Load Test | ■ Define measurable goals for your load test  
■ Gather preliminary information before load testing your system  
■ Organize system information effectively  
■ Use gathered information to plan load tests |
| Module 5—Installing LR Components | ■ Determine where to install the LR components  
■ Identify the software and hardware required for installation  
■ Troubleshoot LG connectivity issues  
■ Configure the Network Virtualization (NV) settings |
| Module 6—Introduction to Scenarios | ■ Describe the elements of a LR scenario  
■ Explain the basic steps for creating a scenario  
■ Configure a LG in the scenario |
| Module 7—Using Run-Time Settings | ■ Define run-time settings for load testing  
■ Describe the difference between run-time settings for scripts and scenarios  
■ Configure run-time settings based on load testing goals |
| Module 8—Scheduling Scenarios    | ■ Configure scenario scheduling  
■ Describe the scheduling by scenario and by group options  
■ Describe the real-world schedule and basic schedule run modes  
■ Manage schedules using the actions grid and the scenario interactive graph  
■ Stop a scenario |
| Module 9—Using Performance Monitors | ■ Describe the value of performance monitors  
■ Select performance monitors to achieve load test goals  
■ Add measurements for performance-based goals  
■ Prepare for a scenario run  
■ Define the scenario running process |

*Continued on the next page*
## Modules and Objectives

### Module 10—Running a Scenario
- Identify the best practices for running a scenario efficiently
- Run a scenario
- Discuss common run-time errors

### Module 11—Using the Analysis Tool
- Describe the analysis tool and the categories of analysis graphs
- Describe the value of analyzing results
- Use graphs to display data
- Describe graph data and raw data
- Apply granularity to many graphs, merge graphs, and perform auto-correlation
- Use service-level agreement (SLA) reports
- Run report generation utilities

---

### Virtual User Generator (VuGen)

#### Module 1—Course Overview of Virtual User Generator 12.0 Essentials
- Identify the contents and objectives of the course
- Define the class schedule and class logistics
- Identify the related courses
- Discuss the lab environment details

#### Module 2—Introduction to Virtual User Generator (VuGen) 12.0
- Define Virtual User (Vuser) and VuGen
- Explore the new look and feel of the VuGen UI
- Explain how to use the step navigator and solution explorer
- Discuss the script workflow
- Create scripts using VuGen
- Record business processes with VuGen

#### Module 3—What’s New in VuGen in 12.0?
- Explore the new look and feel of the VuGen UI
- Describe the user experience and protocol enhancements
- Build a community and describe the concept of flexible delivery

#### Module 4—Recording Scripts for Web Applications
- Access and use the protocol advisor for recording
- Create VuGen scripts by recording user steps
- Save scripts in Micro Focus Application Lifecycle Management (ALM)
- Create a Vuser script template
- Create business process reports

#### Module 5—Replaying the Vuser Scripts
- Recognize the debugging tools available in VuGen
- Identify and configure the appropriate web run-time settings for replaying scripts
- Replay and debug the script in VuGen

#### Module 6—Inserting Transactions into the Script
- Explain the use of transactions in a script
- Add a transaction into a script during recording
- Insert a transaction into a script after recording

#### Module 7—Parameterizing a Script
- Define parameterization
- Determine when to parameterize a script
- Create and modify parameter lists
- Create new parameters
- Work with parameter properties
- Configure run-time settings for parameters

#### Module 8—Verifying the Vuser Scripts
- Identify the need to use verification in scripts
- Define verification for scripts
- Add text checkpoints during and after recording of scripts

*Continued on the next page*
### Modules

<table>
<thead>
<tr>
<th>Modules</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module 9—Creating Actions for a Web Script</strong></td>
<td>Define an action for a web script</td>
</tr>
<tr>
<td></td>
<td>Record a script with multiple actions</td>
</tr>
<tr>
<td></td>
<td>Create action blocks</td>
</tr>
<tr>
<td><strong>Module 10—Using the VuGen Editor</strong></td>
<td>Work with the VuGen Editor</td>
</tr>
<tr>
<td></td>
<td>Send customized output messages to the Replay Log</td>
</tr>
<tr>
<td></td>
<td>Identify basic C code, including statements, variables, and functions</td>
</tr>
<tr>
<td></td>
<td>Apply basic debugging techniques in VuGen</td>
</tr>
<tr>
<td><strong>Module 11—Using the Advanced Scripting Techniques for Vuser Scripts</strong></td>
<td>Explain the general LR functions</td>
</tr>
<tr>
<td></td>
<td>Explain the protocol-specific functions</td>
</tr>
<tr>
<td><strong>Module 12—Auto Correlation after Recording</strong></td>
<td>Describe correlation</td>
</tr>
<tr>
<td></td>
<td>Work with the Correlation Studio</td>
</tr>
<tr>
<td></td>
<td>Correlate dynamic values after recording a script</td>
</tr>
<tr>
<td><strong>Module 13—Using Manual Correlation</strong></td>
<td>Explain the manual correlation process</td>
</tr>
<tr>
<td></td>
<td>Configure the parameters for correlation</td>
</tr>
<tr>
<td></td>
<td>Correlate a script manually by:</td>
</tr>
<tr>
<td></td>
<td>– Using the WDiffutility</td>
</tr>
<tr>
<td></td>
<td>– Adding the web_reg_save_param_excorrelation function</td>
</tr>
<tr>
<td></td>
<td>– Parameterizing the dynamic value in the script</td>
</tr>
<tr>
<td><strong>Module 14—Auto Correlation during Recording</strong></td>
<td>Create correlation rules to auto correlate during recording</td>
</tr>
<tr>
<td></td>
<td>Regenerate and record scripts</td>
</tr>
<tr>
<td></td>
<td>Import and export correlation rules</td>
</tr>
</tbody>
</table>

### About Micro Focus

Micro Focus is a global software company with 40 years of experience in delivering and supporting enterprise software solutions that help customers innovate faster with lower risk. By applying proven expertise in software and security, we enable customers to utilize new technology solutions while maximizing the value of their investments in critical IT infrastructure and business applications. As a result, they can build, operate and secure the IT systems that bring together current business logic and applications with emerging technologies—in essence, bridging the old and the new—to meet their increasingly complex business demands.