



Databridge Client Console

Version 6.6 Service Pack 1

Legal Notices

© Copyright 2020 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Patents

This Micro Focus software is protected by the following U.S. patents: 6983315, 7571180, 7836493, 8332489, and 8214884

Trademarks

Micro Focus, the Micro Focus logo, and Reflection among others, are trademarks or registered trademarks of Micro Focus or its subsidiaries or affiliated companies in the United Kingdom, United States and other countries. RSA Secured and the RSA Secured logo are registered trademark of RSA Security Inc. All other trademarks, trade names, or company names referenced herein are used for identification only and are the property of their respective owners.

Third-Party Notices

Third-party notices, including copyrights and software license texts, can be found in a 'thirdpartynotices' file located in the root directory of the software.

Contents

Introducing the Client Console	5
1 Tour the Client Console	7
Tour the Client Configurator	9
Descriptions of the Panes	11
Icon Descriptions	12
2 Getting Started	15
Connect to the Service	15
Add a Data Source	16
Set Service Properties	17
Switching from Command-Line to Service Operations	18
Exclude Data Sets from Cloning and Tracking	20
Define/Redefine and Generate Scripts Commands	21
Clone a Data Source or Track Updates	22
3 Customize the Client Configuration	23
Flatten OCCURS Items	23
Handle DMSII Reorganizations	24
Client Configuration Dialog Box	26
Bulk Loader Parameters	26
Customizing Parameters	28
Logging/Tracing Parameters	38
Processing Parameters	40
4 Customize a Data Source	51
DMSII vs. Relational Database Terminology	52
Search for DMS Items	52
Cloning a Number as Alpha	53
Cloning an Alpha item as a Number	54
Cloning an Alpha item as Binary	54
Combine DMS Items	54
Split a DMS Item	56
Cloning an Item as a Date Type	56
Cloning an Item as a Date/Time Type	57
Custom Format Dates	57
Cloning an Alpha(36) item as a Uniqueidentifier	58
Customization Options for REAL Item	58
Split Variable Format Data Sets	59
Add a User Column	59
Rename a Table or Column	60

Change the SQL Data Type	61
Create Composite Keys	62
5 Working with Data Sources	63
Best Practices	63
Set Up Client Log Files	63
Show or Hide Data Sets	64
View Multiple Data Sources	64
Disabling, Unloading, or Removing a Data Source	65
Change the SOURCE Password	65
Change the Database Password	66
6 Security Options	67
Manage User Permissions	67
Manage Password Security	68
Set Default Password	69
Change the Console Password	69
7 Managing Operations	71
Stop a Process	71
Schedule Updates	71
View Audit Information	72
Get Statistical Information	72
Set Up Service Log Files	73
Override the Databridge Engine COMMIT Frequency	74
Resynchronize the Service	74
Default Trace and Log Options Dialog Box	75
8 Backing Up the Configuration	79
Archive User Scripts	79
Export the Client Configuration to a File	80
Import a Client Configuration File	80
Display the Client Control Tables	81
Key to Control Table Columns	81

Introducing the Client Console

From the Client Console, you can manage all Client operations, from customizing the layout of your relational database to cloning data and scheduling updates. In addition, the Client Console provides a single window from which you can monitor multiple data sources, view client output messages and performance statistics, and update your settings. For a description of the graphical user interface, see [“Tour the Client Console” on page 7](#).

The Client Console uses the Eclipse integrated development environment. To learn about the Eclipse open source community and the features available to the Client Console, see www.eclipse.com (<http://www.eclipse.com>).

Tutorials

You can find instructions for getting started with the Client Console in this guide and in tutorials that are available from the Help menu in this application.

Reference

See the *Databridge Client Administrator's Guide* for a complete reference to the Databridge Client, including configuration settings, user scripts, and command-line Client operations. This guide is included with the Databridge software as a PDF file (`client-admin.pdf`).

1 Tour the Client Console

The Client Console is a GUI application that allows you to view and manage your data sources. You can accomplish each phase of the replication process by using menu commands and options.

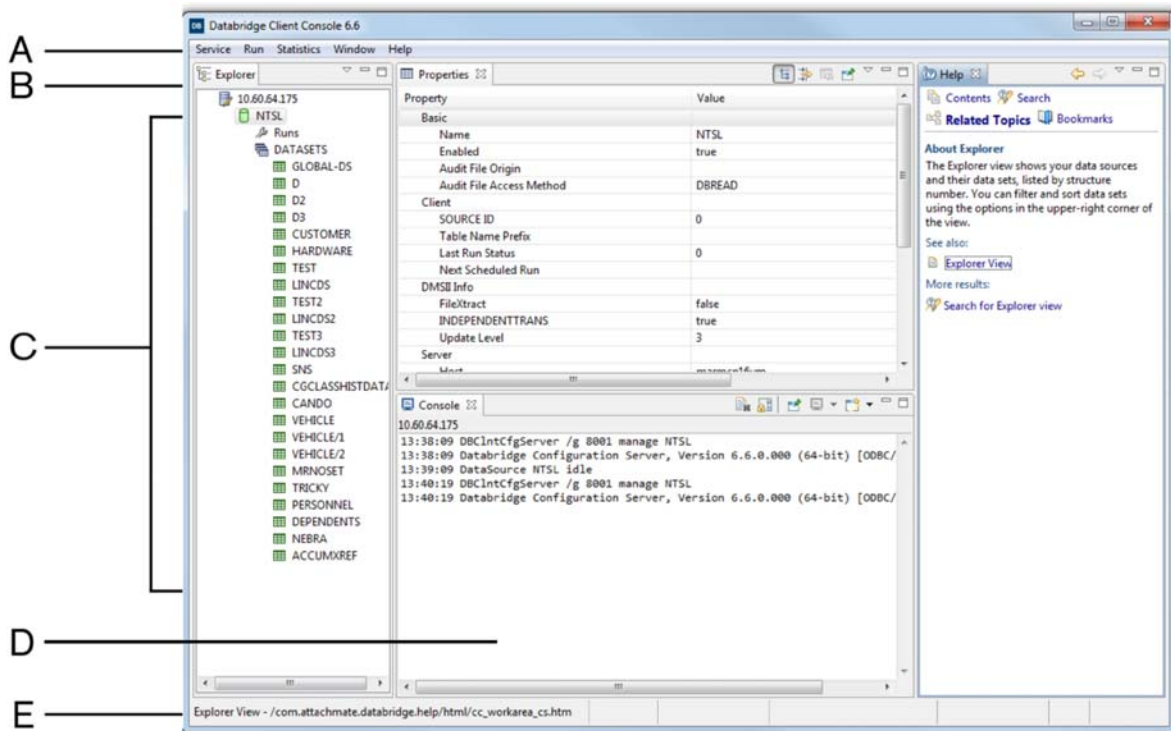


Figure 1 - Client Console Work Area

A. Menu bar B. Title tab for “Explorer” pane C. Data source icons D. “Console” pane E. Status line

Client Console Work Area

The Client Console work area (Figure 1) is divided into multiple panes that display different types of information; for example, **Explorer**, **Console**, **Properties**, and **Help**. You can arrange the panes within the application window by clicking a pane’s title tab (Figure 1-B) and dragging it to a new position.

Menu Bar

The **menu bar** (Figure 1-A) is context-sensitive, displaying different menus depending on which pane you are working in, and sometimes depending on which component (e.g. data source, data set, or service) you’ve selected within the pane. You can also access many of the menu items listed in the menu bar by right-clicking a component within a pane to open a pop-up or context menu.

Explorer Pane

The **Explorer** pane displays a tree of the service nodes and their associated data sources that are maintained by a connected Databridge Client Manager service. Data sets are listed by structure number. You can filter and sort data sets using options from the **Explorer** pane menu (Figure 2-C).

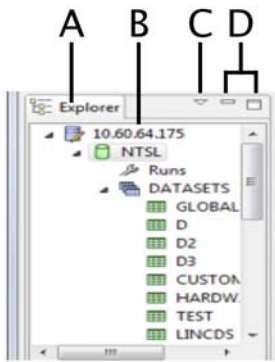


Figure 2 - Explorer Pane

A. Explorer pane tab B. Selected service C. Explorer pane menu D. Minimize & Maximize controls

Within the **Explorer** pane (Figure 2), you can select a service, a data source, or any of its related items in the tree view by clicking on it (you can tell which item you've currently selected by looking for the blue highlight, Figure 2-B) to activate items on the **menu bar** specific to data sources, e.g. the **Data Source** menu (Figure 1-A). Alternatively, you can right-click a service, a data source, or any of its related items to open a pop-up or context menu (Figure 3).

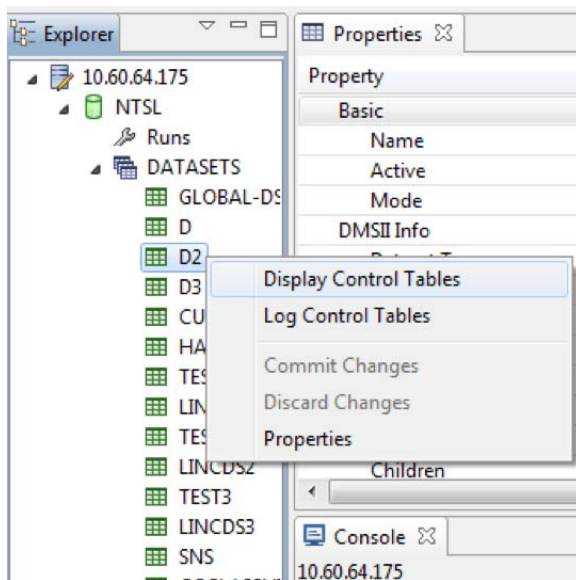


Figure 3 - Explorer Pane pop-up menu

Console Pane

The **Console** pane (Figure 4) displays Client-related activity for each data source, including log output, exit codes, and commands. (This information is the equivalent of the program output displayed when running the command line Client.)

You can open dedicated **Console** panes for each individual data source by using the Open Console control (Figure 4-E). The Pin Console control (Figure 4-C) lets you keep a particular console on top.



Figure 4 - Console controls

A. Clear Console B. Scroll Lock C. Pin Console D. Display Selected Console E. Open Console

Status Line

At the lower edge of the Client Console window, the **status line** (Figure 1-E) shows the status of the connection to the currently selected service and currently active run and whether additional user input is required to complete a task.

When a `process` command is running, additional fields display audit file processing information: the Audit File Number (AFN), the Audit Block Serial Number (ABSN), a progress bar for the designated audit file, the current DMS II AFN, and lag time. Lag time is the elapsed time between when an update is applied in the DMSII database and when it appears in the relational database.

If multiple data sources are processing at the same time, the **status line** will cycle through each data source status in five-second intervals.

When you close the Client Console, all information is lost except the services to which the console is connected, whether to automatically connect to a given service at startup, and the layout of the Console window.

Tour the Client Configurator

The Client Configurator work area looks much like the Client Console work area. The **menu bar** (Figure 1-A) includes many of the same options that are available from context menus (that is, by right-clicking an item in the **DMSII** or **Relational** pane). The **toolbar** (Figure 1-B) includes options to save, discard, and view your changes. The **status line** (Figure 1-F) shows the status of the connection to the currently selected database and whether additional user input is required to complete a task.

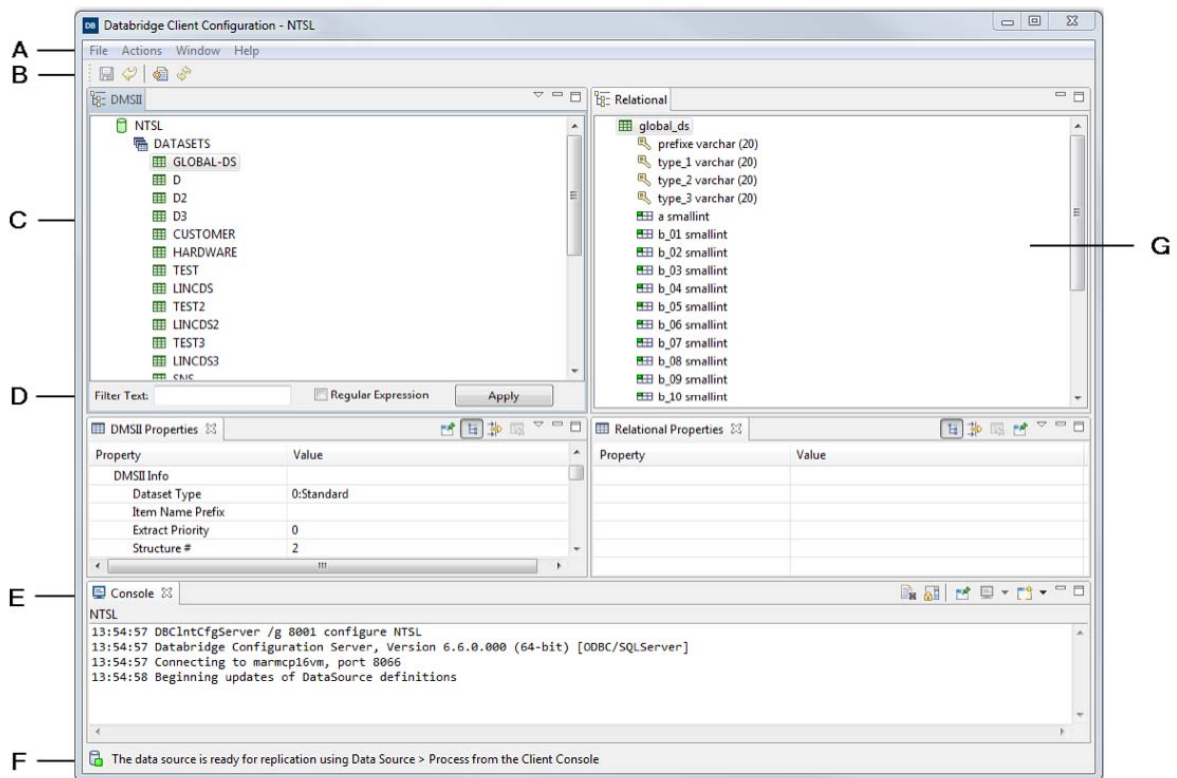


Figure 1 - Client Configurator Work Area

- A. Menu bar B. Toolbar C. DMSII database pane D. Filter Text box (search)
 E. Console pane F. Status Line G. Relational database pane

The top left two panes of the work area display the DMSII environment; the top right two panes of the work area display the relational database environment.

The **DMSII database** pane (Figure 1-C) shows the data source and its data sets and their items. If you right-click an item in this pane, you can open a context menu and set customization options and modify how a DMSII item will be mapped to the relational database. From the **DMSII Properties** pane, you can supply additional parameters (e.g. the format for a date).

Use the **Filter Text** box (Figure 1-D) to quickly find DMS items you want to modify or select for replication. The filter options use a wild-card pattern matching method to show DMS items with specified characteristics. For example, if you type "-DT", only items whose names contain the character string "-DT" (e.g. "BIRTH-DT" and "ELEVEN-DT-TWELVE") will be displayed.

The right side of the window (**Relational database** pane, Figure 1-G) shows data tables and columns that correspond to the DMS items selected for replication. This pane reflects your changes and shows how those items will appear in the relational database. The **Relational Properties** pane includes additional customization options.

NOTE: The **Relational database** panes do not feature context (right-click) menus.

The **Console** pane (Figure 1-E) allows you to monitor the DBCIntCfgServer program output while you're working in the Client Configurator.

At the lower edge of the Client Configurator window, the **status line** (Figure 1-F) shows the status of the connection to the service and the data source and whether additional user input is required to complete a task.

For information about using the Client Configurator, see “[Customize a Data Source](#)” on page 51.


Descriptions of the Panes

The following table includes a description of each of the Client Console panes (in alphabetical order). Panes appear in the Client Console and the Client Configurator window unless otherwise noted.






Pane	Description
Console	<p>Displays the screen output of a client running in the background.</p> <p>NOTE: Each client has its own console pane.</p>
Explorer	<p>A tree-view representation of Client Manager services and data sources.</p>
DMSII and DMSII Properties	<p>(Client Configurator only) The DMSII pane in the Client Configurator shows configurable data sets for the data source you are configuring.</p> <p>The DMSII Properties pane shows the properties for most items selected in the DMSII pane, specifically the data source, data set, and DMS item.</p> <p>You can customize items by using the context menu (right-click on the item to get the menu that is tailored for the item in question) You can modify properties of the selected item in the DMSII Properties pane that appears below the DMSII pane. (Click the Value column to show an input field, a list box, or a switchable value.)</p> <p>The controls in the upper-right corner of the DMSII Properties pane change the way properties are displayed within the pane.</p>
DMSII Counters	<p>(Client Console only) Shows counters based on incremental statistics that the client accumulates while processing audit files. These are shown after the current audit file is processed.</p> <p>At the bottom of the chart, AFN and ABSN data identifies the audit file and the audit block that are currently being processed. A scroll bar lets you navigate counters for previously processed audit files. Only non-zero values are graphed in the chart. The AFN and ABSN data at the bottom of the chart identifies the data set.</p>
Lag Time	<p>(Client Console only) The elapsed time between when an update is applied in the DMSII database and when the update appears in the relational database. The plotted value is the difference between the current time and the approximate timestamp of the update in the audit file.</p> <p>From the Lag Time pane menu, click Client Status Options to modify the number of charted data points and the frequency with which the Client is queried for status updates. Data is saved on a per-run basis. Save the statistics used to generate the chart into a CSV (comma delimited) file by clicking Export Statistics. To see the statistics for another run, select that run in the Explorer pane.</p>

Pane	Description
Performance Statistics	(Client Console only) Shows raws statistics based on unsolicited messages the client sends at the end of audit files. The Client log pane shows edited forms of these statistics.
Properties	(Client Console only) Shows the properties of an item selected in the Explorer pane. The type of item that's selected determines the properties, values, and options that appear in this pane.
Relational and Relational Properties	<p>(Client Configurator only) The Relational pane shows the data tables and columns as they are mapped in the relational database.</p> <p>The Relational Properties pane shows properties for the selected items. From this pane, you can modify data types and table and column names.</p> <p>NOTE: To customize the layout of the relational database, use the DMSII Properties pane.</p>
Task Time	(Client Console only) The AFN and ABSN data at the bottom of the chart identifies the audit file and the audit block that are currently being processed. A pie chart represents the percentage of time spent on each task the client performed when processing the current audit file.






Icon Descriptions

When an icon includes a pencil, such as this service icon  , tracing is enabled for that item.

Data Source Icons

-  Active data source
-  Disabled data source
-  Data source with unsaved changes or a new data source that hasn't been defined or customized
-  Data source that requires attention
-  Data source is offline because it's being customized in this or another Client Console. The data source will be available when the Client Configurator session ends.

Data Set Icons

-  Data sets
-  Inactive data set
-  Active data set
-  Data set that requires attention
-  Modified data set (unsaved changes)


DMSII Item Icons


-  DMSII item
-  Modified DMSII item


Data Source Icons


 DMSII group

 Key


 Reorganization indicator of deleted DMSII item


 Modified DMSII group


 User column


 Reorganization indicator of added DMSII item

Client Run Icons

 Default client run icon

 Failed client run

 Active client run

 Completed client run

2 Getting Started

This section provides instructions for connecting to the Client Manager service on the Client system and adding data sources.

NOTE: Throughout this documentation, the term *service* refers to both the service (Windows) and the daemon (UNIX). The term *Databridge Server* refers to both Databridge Server (DBServer) on the host and Databridge Enterprise Server (DBEnterprise) on a Windows computer.

Connect to the Service

Use this procedure to connect to the Client Manager service installed with the Databridge Client software on the Client system. If you have multiple clients on same machine, you can manage them with a single service. The service controls all client runs and the DBCIntCfgServer program. The DBCIntCfgServer program acts as a server and performs all operations (except for process and clone commands) initiated by the Client Console and the Client Configurator.

After you add a data source to the service, you can schedule regularly occurring client runs. For information, see [“Schedule Updates” on page 71](#).

To add a service

NOTE: If you upgraded your Databridge Client software and user scripts from a previous version, skip this task and the next, [“Add a Data Source” on page 16](#). The service and your data sources should automatically appear in the Client Console.

- 1 Make sure that the Client Manager service is running on the Client system. (For instructions, see the *Databridge Installation Guide*.)
- 2 From the **Service** menu, click **Add Service**.
- 3 In the **Add Service** dialog box, type the host name or IP address and port number, and click **OK**. The default port number for the service is 8001.
- 4 In the **Connect to** dialog box, type your user ID and a password, and click **OK**.
The next time you start the Client Console, it automatically connects to the configured service. To change this default, select the service in the **Explorer** pane, and in the **Properties** pane, deselect **Connect on Startup**.

To remove a service

- ♦ In the **Explorer** pane, right-click the service and choose **Disconnect**.
- ♦ Right-click the service again and choose **Remove Service**.


Add a Data Source

Use this procedure to specify data sources for replication. You can add multiple data sources to a service.

If you've been using the command-line Client, you can add your existing data sources to the Client Console. See [Switch from Command-Line to Service Operations \(page 18\)](#).

NOTE: If you use Databridge Enterprise Server, you can add data sources that have been defined in that program. For the host parameters (step 6), use the host and port numbers for Enterprise Server.

To add a data source

- 1 Make sure that your Databridge Server is running and the relational database is configured in accordance with the manufacturer's instructions. SQL Server databases must have an ODBC data source created using the Control Panel before you can continue.
- 2 Make sure that the connection to the service has been established.
- 3 In the **Explorer** pane, right-click the service , and do one of the following:
 - ◆ To add a new data source, choose **Add Data Source** in the context menu.
 - ◆ To add an existing data source whose working directory is a sub-directory of the service's working directory and is named using the lowercase form of the data source name, choose **Add Existing Data Source** in the context menu. Type the data source name and click **OK**. Skip the remaining steps.
- 4 In the **Add Data Source** dialog box, for **Name**, type the name that exactly matches the SOURCE entry in the corresponding Databridge Server configuration. The client will use this name as the data source name.
- 5 Do one of the following:
 - ◆ For SQL Server, in the **ODBC data source** box, type the name of the ODBC data source you created in step 1.
 - ◆ For Oracle, type the database name.
- 6 Specify the type of authentication:
 - ◆ For SQL Server, select Integrated Windows authentication or SQL Server authentication. If you select SQL Server authentication, type the SQL login ID and password.
 - ◆ For Oracle, type a login ID and password.
- 7 Enter the host parameters:
 - ◆ For **Hostname**, type the DNS name or IP address of the host.
 - ◆ For **Port**, type the TCP/IP port number assigned to the Databridge Server.

8 For the **Post-add actions**, do one of the following:

- ◆ Select **Customize Data Source** to modify the configuration parameters in the **Client Configuration** dialog box using the Client Configurator and create the data source. For more information, see [“Customize the Client Configuration” on page 23](#).
- ◆ Select **Define and Generate Scripts** to use the default configuration parameters and prepare the data source for cloning. **Define** uses the default parameters to create the data source, but requires that you manually generate scripts. If there are user scripts present in the user scripts directory, they will be used to customize a new data source, but not an existing data source.

9 Click **Finish**.

If the data source parameters are valid, the data source appears in the Explorer pane below the service. The service adds the specified login ID and password to the client configuration file (`dbridge.cfg`).

If the data source parameters are invalid, an error will be displayed at the top of the dialog box. Correct your entries and try again by clicking **Finish**.

To remove a data source

This command deletes all of the cloned data of a data source and optionally removes all related files.

- 1 In the **Explorer** pane, right-click the data source and click **Advanced > Remove Data Source** in the context menu.
- 2 In the warning dialog box, select the check box if you want to delete all files and directories related to the data source, and then click **OK**.

Set Service Properties

Use the following procedure to set service parameters for all data sources associated with a service.

Additional service parameters, such as scheduling, are available in the **Client Configuration** dialog box. However, these parameters affect only the selected data source. See [“Customize the Client Configuration” on page 23](#).

To set service parameters in the Properties pane

- 1 From the **Explorer** pane, right-click the service and select **Properties** in the context menu.
- 2 In the **Properties** pane, specify the following:

Service

IPC Port

The TCP/IP port number on which the Client Manager service listens for connection requests from the Client Console or client runs. This value reflects the port number you used when you added the service. By default, this number is 8001.

Legacy parameter name is `ipc_port`.

NOTE: If you change the port number you must restart the service before the change will take effect. Disconnect the console before you do that.

Number of Script Launcher Threads	<p>Defines the size of the pool of threads that the service uses to launch end of run and externally launched scripts. The reason for having threads is that the service waits for the scripts to complete. Without threads long scripts would cause the service to become unresponsive. If all threads are busy, the execution of the scripts is delayed until a thread becomes available.</p> <p>Legacy parameter name is <code>n_scr_threads</code>.</p>
Startup Delay (seconds)	<p>Defines the delay that the service imposes when starting consecutive process command runs for various data sources. This is designed to alleviate the bottleneck in the Databridge Server that results and causes some of the command to fail.</p> <p>Legacy parameter name is <code>startup_delay</code>.</p>
Enable Status File	<p>This parameter, which only applies to clustered Windows systems, indicates that the service should maintain a status file that reflects the state of the data sources. Should the service restart as a result of a node switch, all process command runs that were previously active will then automatically restart.</p>
Session Startup Timeout (seconds)	<p>This parameter specifies the length of time that the service waits for input from a new connection before forcing a disconnect. This parameter is designed to protect against a flood of rogue connection requests that would otherwise cripple the service. In some cases, the default value of 2 seconds might be too low. This parameter allows you to adjust the value to best suit your environment.</p>

Switching from Command-Line to Service Operations

Use this procedure if you currently run the Databridge Client from the command line and want the Client Manager service to run it, or if you need to add existing data sources to the service.

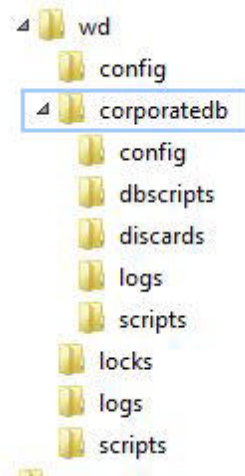
To operate the service on the Client machine, you'll need a specific directory structure, referred to as the *service's working directory*.

CAUTION: Important! If you're in the process of upgrading your Databridge software, use the instructions in the *Databridge Installation Guide* for upgrading the Client.

To switch to the service based client (Windows)

- 1 When installing the software the service's working directory is automatically created, unless it already exists.
- 2 Install the Client Console that matches the version of Databridge Client software you use. For instructions, see the *Databridge Installation Guide*.
- 3 Do one of the following:
 - ♦ If you use a text Client configuration file, go to step 4.
 - ♦ If you use a binary Client configuration file, skip to step 6.
- 4 From the data source directory, locate the `config` folder, and copy the text configuration file `dbridge.cfg` to a file named `dbridge.ini`.

The following image shows an example of a working directory for a data source named "corporatedb".



- 5 From a command prompt, set the current directory to the working directory for your data source using a `cd` command, and then run the `dbutility IMPORT` command. This creates a binary configuration file required by the service.
- 6 Make sure that the Client Manager service is running. From the **Start** menu, click **Control Panel > Administrative Tools**, double-click **Services**, double-click **Databridge Client Manager [version]**, then click **Start**.
- 7 Start the Client Console from the **Start** menu by clicking **All Programs > Micro Focus Databridge [version] > Databridge Client Console [version]**.
- 8 Add a service, if you haven't already done so. For instructions, see [Connect to the Service](#).
- 9 With the service selected, from the **DataSource** menu, click **Advanced > Add Existing Data Source**, type the data source name (for example, `corporatedb`), and then click **OK**.
Or, right-click the service in the **Explorer** pane, click **Add Existing Data Source** from the context menu, type the data source name (for example, `corporatedb`), and then click **OK**.

To switch to the daemon based client (UNIX/Linux)

- 1 Log on as the `userid` specified in the `USERID` parameter of the file `globalprofile.ini`. This is the same `userid` you currently use to run the command-line client.
- 2 Set the current directory to the home or other directory to which you have write access and copy the script file `dbdaemon` from the install directory to it. This allows you to make changes to the script file.
- 3 In an editor, open the script file `dbdaemon`.
- 4 Make sure that the environment variables (such as `INSTALLDIR`, `WORKING_DIR`, `ORACLE_HOME`) are correct for your system and edit them as needed.
- 5 Save and close the script file.
- 6 Start the daemon by typing the following:

```
dbdaemon start
```
- 7 To verify that the daemon is running use the `ps` command, which should produce output similar to the following:

```
dbadmin@VMOPENSUS114-64:~> ps -ef | grep DBC
```

```
dbadmin
1110      1  0 12:00 ?          00:00:00
/opt/dbridge66/DBCIntControl
```

- 8 If the daemon doesn't start, make sure that the `WORKING_DIR` and `INSTALLDIR` environment variables are correct in the script `dbdaemon`. Also, check the service's working directory to determine if the file `dbdaemon.log` was created and if it contains any clues.
- 9 Start the Client Console from the **Start** menu by clicking **All Programs > Micro Focus Databridge [version] > Databridge Client Console [version]**.
- 10 Add a service, if you haven't already done so. For instructions, see [Connect to the Service](#).
- 11 With the service selected, from the **DataSource** menu, select **Advanced > Add Existing Data Source**, type the data source name (for example, `corporatedb`), and then click **OK**.
Or, right-click the service in the **Explorer** pane, click **Add Existing Data Source** from the context menu, type the data source name (for example, `corporatedb`), and then click **OK**.
- 12 Type `su` to switch to the root user and then copy the script `dbdaemon` from the location you specified in step 5 to the following location:
 - ♦ (Linux/Solaris) `/etc/init.d`
 - ♦ (AIX) `/etc/rc.d/init.d`
 - ♦ (HP-UX only) `/sbin/init.d`
- 13 To make the operating system automatically start the daemon whenever the system starts, consult the documentation for your operating system.

Exclude Data Sets from Cloning and Tracking

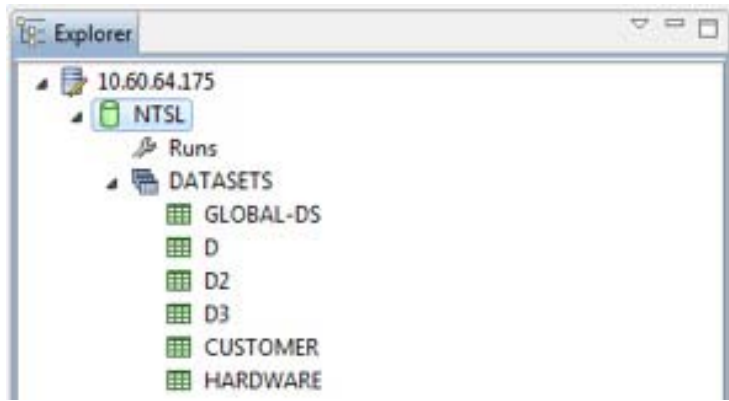
The **Active** check box in the Client Console determines whether data sets are included in cloning and tracking processes. You can temporarily stop tracking for a data set by clearing the **Active** check box in the Client Console.

CAUTION: If you deselect the **Active** check box for a data set in the Client Configurator, tables that are mapped from the data set are removed from the `DATATABLES` and `DATAITEMS` client control tables. Any customizations you made to these tables in the **Relational** pane are also lost.

If you want to filter out data sets that contain confidential or sensitive information, consider using a logical database or a filtering routine in the `DBGenFormat` utility on the host. The Client will have no record that these data sets exist.

To exclude a data set

- 1 From the **Explorer** pane, click the triangle pointing to the data source containing a data set you wish to exclude to expand the data source node and show the data sets.



- 2 Right-click a data set and click **Properties** from the context menu.
- 3 In the **Properties** pane, uncheck the **Active** check box.
The item becomes outlined in red, indicating that it has been modified.
- 4 To save the change, right-click the data set (or the DATASETS node when you have multiple changes) and select **Commit Changes** from the context menu.
The icons return to normal. If a data source contains active and inactive data sets, the **Active** value in the **Properties** pane is <multiple>.
- 5 (Optional) To show inactive data sets in the **Explorer** pane, use the filters from the **Explorer** pane menu. Inactive data sets are hidden by default.

Define/Redefine and Generate Scripts Commands

You may need to select the **Define/Redefine** and **Generate Scripts** commands if you create a new data source without customizing it (that is, you selected no post-add action in the **Add Data Source** dialog box).

Define/Redefine

The **Define/Redefine** command reads the DMSII information from the data source and starts building the relational database tables, layouts, and control information for the data source. If you run the **Define/Redefine** command after the relational database is created, it determines what has changed on the DMSII side and attempts to match it in the relational database while preserving the existing information (this is equivalent to a command-line Client `redefine` command).

To define a data source

- ♦ In the **Explorer** pane, right-click the data source and choose **Advanced > Define/Redefine** from the context menu.

Generate Scripts

The **Generate Scripts** command creates a set of script files and puts them in the `dbscripts` subdirectory of the data source's working directory. These script files include SQL scripts the Client uses to create tables and indexes, drop tables, and remove false duplicate records from tables. Also included are command files (shell scripts for UNIX clients) that run bulk loader tasks. (Approximately six script files are created for each data table.)

To generate scripts for a data source

- ◆ In the **Explorer** pane, right-click a data source and choose **Generate Scripts** from the context menu.

Clone a Data Source or Track Updates

Use the **Process** command to clone a data source and track updates. Before you clone a data source, make sure that you've completed all customizations to the data source.

To run a Process command

- ◆ In the **Explorer** pane, right-click the data source and choose **Process** from the context menu.

During the initial clone, DMSII data for the data source is extracted and cloned to the relational database.


After the data source is cloned, you can run the **Process** command anytime you want to update the relational database. The **Process** command tracks DMSII updates by reading the audit trail and synchronizes the received data to mirror the DMSII database. To schedule updates to occur automatically, use the **Scheduling** parameters in the **Client Configuration** dialog box. See [“Customize the Client Configuration” on page 23](#).

3 Customize the Client Configuration

Use this procedure to set the global customization parameters for a data source before you clone it. The **Client Configuration** dialog box includes global customization settings that modify the table layout of the relational database. It also includes settings that manage daily operations, such as scheduling, log creation and filenames, and stop conditions.

To set the customization parameters

NOTE: When you open the **Client Configuration** dialog box from the Client Configurator (versus from the Client Console), you can customize your data source without restrictions. This is suitable for uncloned data sources. After you clone, we recommend that you open the **Client Configuration** dialog box from the Client Console (**Data Source > Client Configuration**). This ensures that you can only access options that don't require recloning.

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. This will start a new run that has its own window. The remainder of these steps apply to the Client Configurator window.
- 2 In the **DMSII** pane, right-click the data source and click **Client Configuration** from the context menu.
- 3 From the left side of the **Client Configuration dialog box**, click a category or click the triangle pointing to a category to show additional parameters for that category.
- 4 For a description of these options, click the **Help** button  in the lower-left corner of the **Client Configuration** dialog box. (If you select a new category of settings, click the **Help** button again.)
- 5 When you're done, click **OK**.

Your selections are saved to the binary client configuration file (`dbridge.cfg`) located in the config subdirectory of the data source's working directory.

Flatten OCCURS Items

An OCCURS clause is a DMSII construct that describes the number of times an item is present or repeats within a data set. Because relational databases don't have an equivalent construct, additional tables are created for these items, which degrades the performance of update processing. In Databridge Client, if you don't flatten the OCCURS clauses, a secondary table is created. Each OCCURS clause and its keys are mapped to a single row in that table.

CAUTION: Make this type of change only before cloning the data source. Otherwise, you will need to reclone.

To handle OCCURS items

- 1 In the **Explorer** pane, right-click a data source and choose **Customize Data Source** from the context menu.
- 2 In the **DMSII** pane, right-click the data source and choose **Client Configuration** from the context menu.
- 3 On the left of the Client Configuration dialog, click the triangle to expand **Customizing**.
- 4 Under **Customizing**, click **Advanced Parameters**.
- 5 Click on an item with an OCCURS clause in the DMSII view and pick one of the choices from the list box next to "**Flatten Occurs**" in the properties pane for the item. These include "Do Not Flatten", "Flatten Within Table", "Flatten Into Secondary Table", "Flatten Into String (Fixed Fmt)" and "Flatten Into String (Variable Fmt)". The last two choices are only present if the item qualifies for being flattened to a string. Only NUMBER(n) and ALPHA(n) items with OCCURS clauses can be flattened to strings. Refer to the [Flattening OCCURS Clauses](#) section of the Client Administrator's Guide for details on Flattening an OCCURS clause to a string.

When "Flatten Within Table" is selected, the Databridge Client maps each occurrence of the item to a separate column in the primary table, the columns are named by appending a suffix of the form "_dd" to the name of the DMSII item after it converted to lower case and is subjected to the standard transformation for object names (e.g., minus signs are changed to underscores). The number of digits in the suffix is the minimum number of digits needed to hold the number of occurrences in the OCCURS clause (e.g. 2 digits for OCCURS 10 TIMES).

When "Flatten Into Secondary Table", the Databridge Client maps each occurrence of the item to a separate column in a secondary table, that also contains a copy of the keys in the primary table.

NOTE: If you have a large number of OCCURS clauses, as an alternative to flattening OCCURS clauses you can enable the **Optimize SQL updates** parameter by clicking on the corresponding checkbox in the Customizing>Advanced Parameters page of the Client Configuration. This option eliminates all redundant updates (where all the columns are unchanged) for a secondary tables.

Handle DMSII Reorganizations

If a DMSII database reorganization changes the layout of records, parallel changes must be applied to the Client database.

You can configure the service to automatically process any DMSII reorganizations that are detected by process commands by using **Automatic Run Options**.

To handle DMSII reorganizations automatically




- 1 In the **Explorer** pane, right-click the data source and click **Client Configuration** from the context menu.
- 2 On the left, expand **Processing**, and click **Scheduling**.
- 3 For **Automatic Run Options**, enable **Redefine data source when DMSII reorganization detected** checkbox by clicking on it.
- 4 To automatically create scripts when required, enable **Generate scripts after an automatic redefine** checkbox by clicking on it.


If a DMSII reorganization is detected, the Client Manager service runs a `redefine` command, which rebuilds the relational database layout for data sets that have been reorganized. All customizations are preserved. The `redefine` command compares the old and new table layouts to determine if any changes have occurred.

- ◆ If the table layouts haven't changed, the service starts a `process` command to resume operations.
- ◆ If the table layouts have changed (for example, data sets added or existing data sets must be re-cloned as a result of the reorganization), scripts must be generated before processing can resume. If you enable the **Generate scripts after an automatic redefine** checkbox by clicking on it, the Client Manager service automatically runs a `generate` command to create these scripts. If you didn't select this option, create the scripts manually by clicking **Data Source > Generate Scripts**.
- ◆ If the **Reorganize** command is required, you'll need to run this command before you can resume processing. Use the Client Configurator to make any additional changes to the table layouts. Examine the scripts that the client created to ensure that they're correct and then click **Data Source > Reorganize**. The **Reorganize** command generates new scripts for creating tables and stored procedures and running the bulk loader. Then, it alters the layout of all tables affected by the reorganization by running the reorg scripts that were created by the `redefine` command. Finally, it refreshes the stored procedures in the relational database to use the new layouts. When the **Reorganize** command is finished, you can resume processing.

To handle reorganized items manually

If a DMSII reorganization requires structural changes to your replicated data source, you will be prompted to address the changes in the Client Configurator. The Client Configurator uses the following icons for a data source to indicate changes that have occurred:

This icon	Indicates that
	The item was affected by a DMSII reorganization.
	Items were added by a DMSII reorganization.
	Items were removed by a DMSII reorganization.


- 1 In the **Explorer** pane, right-click the reorganized data source and choose **Customize Data Source** from the context menu.
After the Client Configurator starts, it runs a `redefine` command. Only items that have affected by the reorganization appear in the **DMSII** pane.
- 2 In the **DMSII** pane, select an affected data set.
- 3 Customize the new data sets or new columns using the available options in the **DMSII Properties** pane or the **Relational Properties** pane. See [“Customize a Data Source” on page 51](#).
- 4 In the toolbar, click **Save** .
- 5 Choose **File > Exit** from the **menu bar**.

Typically, the Client Configurator automatically runs a `Generate` command, which allows you to resume processing. The next time you start the Client Configurator, the icons for these items will appear as normal.

- 6 If you're prompted to review your scripts, do so.
- 7 After you review and approve the scripts, in the **Explorer** pane, right-click the data source and choose **Reorganize** from the context menu. This command generates new scripts for stored procedures and refreshes the stored procedures in the relational database. You can begin processing immediately.

Client Configuration Dialog Box

Use the Client Configuration dialog box to set global parameters for a data source.

For a description of these options, click the **Help** button  in the lower-left corner of the **Client Configuration** dialog box.

Bulk Loader Parameters

Use the following parameters to control the bulk loader (`bcp`) utility for your relational database—Oracle or SQL Server. Use the following parameters to customize the selected data source before you clone it. Configuration file parameters are included in parentheses.

Delimiter (`bcp_delim`)

(SQL Server only) Specify the character or set of character(s) that are used to separate variable-length fields in the bulk load input records.

Enter one or more characters. Use this option if the data contains alpha fields with TAB characters that need to be preserved. (A possible delimiter value in this case would be `"|"` or `"| |"`.)

Code Page (`bcp_code_page`)

For SQL Server, you can use the names `ACP`, `EOM`, `RAW` or a number such as `1252` to represent a code page. For Oracle, the name is a string such as `"WE8ISO8859P1"`. Consult your Oracle documentation for the actual names.

Copied message (`bcp_copied_msg`)

(SQL Server only) If this parameter is not set correctly, the `bcp_utility` reports bulk loader failures even though the bulk loader worked correctly. Enables the `bcp_auditor` utility to determine whether or not a bulk loader was successful in cases where the database language is not English. For example, in German, this parameter is `"Zeilen kopiert"`, but in English, it is `"rows copied"` (default).

Batch size (`bcp_batch_size`)

(SQL Server only) Specifies the `bcp` utility batch size, which is the number of rows per batch of data copied. This allows the `bcp` utility to load a table in several batches instead of in a single operation. Permitted values are `0` or `1000-10000000` (rows per batch). A value of zero causes the `bcp` utility to load the entire group of records in the data file in one batch. Copying all of the rows of a very large table in one batch may require a high number of locks on the Microsoft SQL Server database.

Packet size (`bcp_packet_size`)

(SQL Server only) Defines the network packet size value for the bcp utility (applies to remote servers only). If you have wide tables, setting this parameter to a packet size larger than the default (4096) can speed up loading the data into the table at the expense of system resources. Before you can set the value for this parameter, you must first enable **Packet size** by clicking the check box to the left of this option.

Maximum temporary storage (max_temp_storage)

(Windows only) Specify a value between 10MB and 3GB for the maximum amount of storage the Databridge Client will use for temporary files. Specifying a non-zero value for this parameter activates the segmented bulk load feature. Use the letter M or G after the number to indicate megabytes or gigabytes.

Maximum loader errors (max_errors)

Controls the bulk loader's tolerance to records that are discarded due to data errors. Use this parameter when you have many bulk loader errors. Increasing the maximum error count allows you to gather all the errors in one run rather than finding 10 errors and then having to start over again. Permitted value range is 0-1000.

SQL loader bind size (sqlld_bindsize)

(Oracle only) Defines the value to be used for the BINDSIZE parameter for SQL*Loader operations. Increasing this value can speed up SQL*Loader operations when not using DIRECT mode (for example, running remote to a database on a UNIX system). Use this option only when running dbutility for a remote Oracle database on UNIX or Windows and the inhibit_direct_mode parameter is enabled.

SQL loader direct (inhibit_direct_mode)

(Oracle only) Enable this parameter if the Databridge Client accesses a remote Oracle database using SQL*Net between two dissimilar architectures (for example, Windows and UNIX). This inhibits the use of the direct=true option when invoking SQL*Loader in the command files so that you do not have to remove the string direct=true from every call on SQL*Loader. When you enable this parameter, we recommend that you increase the **SQL loader bind size** for better performance. Leave this parameter disabled if your Oracle database is on the same machine as the Databridge Client.

SQL loader parallel (enable_parallel_mode)

(Oracle only) This parameter, which is only meaningful when DIRECT mode is enabled, causes the generate command to add the specification parallel = true to the SQL*Loader command line. Parallel mode makes the SQL*Loader run faster at the expense of additional system resources. Applies only to Oracle.

SQL loader rows (sqlld_rows)

(Oracle only) Defines the value to be used for the ROWS specification for SQL*Loader operations. Use this option only when running dbutility for a remote Oracle database on UNIX or Windows and the inhibit_direct_mode parameter is enabled.

Verify load (verify_bulk_load)

Specifies how the Databridge Client will handle the results of the bulk copies.

When **Don't verify** is selected, no action is taken. When **Verify** (default) is selected, after the final bulk load completes, the Databridge Client gets the record count of each table and compares it to the total number of records passed to the bulk loader. If the two counts differ, the Databridge Client displays a warning message. **Verify, exit on error** is the same as **Verify** except that the Client terminates so that **you can investigate the reason for the mismatch**.

Customizing Parameters

Use the following parameters to customize the selected data source before you clone it. Configuration file parameters are included in parentheses.

General

Allow NULLs (allow_nulls)

Stores DMSII null data and data items that contain bad values (excluding keys) as NULLs in the relational database.

Set blank columns to NULL (set_blanks_to_null)

Stores zero-length character data (that is, "") as NULL instead of a single space. This only applies to columns that are not part of the index.

Allow NULL dates (use_nullable_dates)

Allows the Client to treat a single MISER date in the index as nullable.

Use brackets around table names that are reserved words (bracket_tabnames)

Allows the SQL Server Client to use table names that are reserved words by enclosing them in square brackets.

Ignore new data sets (suppress_new_datasets)

Indicates whether or not the program maps new data sets created during a DMSII reorganization. If this parameter is set to True, the active columns for new data sets are set to 0. If you later decide that you want to replicate these data sets, enable the **Active** checkbox in the data set's properties and then run a `redefine` command, unless you are using the Client Configurator.

Use column prefixes (use_column_prefixes)

Extends the table name prefix specified in the DATASOURCES client control table to all column names. This prefix allows you to distinguish between identically named data sets in multiple data sources. If a prefix isn't defined, this setting has no effect.

Strip Data Set prefixes (strip_ds_prefixes)

This parameter makes the `define` and `redefine` commands set the `item_name_prefix` column in the DATASETS table to the data set name. This is useful when all DMSII data item names use a common prefix; for example, using the data set name followed by a dash. This parameter provides a quick way of stripping those common prefixes without writing any user scripts or using the Client Configurator (as renaming every column requires a lot of work). If the prefix is an abbreviated form of the data set name (e.g. SVHIST instead of SV-HISTORY), set the value of the data set's `item_name_prefix` column in the properties pane after clicking on the data set name in the DMSII view of the Client Configurator.

DMSII related parameters

Enable DMSII links (enable_dms_links)

Allows Databridge to implement DMSII links. DMSII links are implemented using AA values as foreign keys. To use this parameter, you must set the DBEngine Links parameter to True. For more information, see the *Databridge Host Administrator's Guide*.

Track Variable Format data sets (track_vfds_nolinks)

Tracks variable-format data sets that contain links. The links themselves are not tracked or updated. When a record is created in a variable-format data set, links are set to NULL. If the application assigns the links to point to other records, the client database will not contain these new link values until the variable-format data set is recloned. This parameter is enabled by default.

When this parameter is disabled, variable-format data sets are set to mode 11 after initial cloning and not updated.

Extract embedded data sets (extract_embedded)

Embedded data sets are not normally handled by the Databridge Engine when INDEPENDENTTRANS is False, as the client cannot apply fixups or updates. When this parameter is enabled, the Client enables a DBEngine parameter that causes the DBEngine to go through the data extraction process for embedded data sets that cannot be tracked and updated. When this parameter is disabled, Databridge Client ignores embedded data sets. If INDEPENDENTTRANS is True, the Databridge Client can clone and update embedded data sets regardless of the setting of this parameter.

Read NULL record values (read_null_records)

Requests NULL VALUES for data set records from DBEngine during the define/redefine commands. NULL VALUES are stored in a binary file (*datasource_NullRec.dat*) and are retrieved at the beginning of a `process` or a `clone` command. If this parameter is disabled, numeric data items whose bits are all 1 (high values) and character data items whose bits are all 0 (low values) are considered to be NULL. When this parameter is enabled, testing for NULL is more accurate.

MISER database (miser_database)

Enable this parameter when running Databridge at a MISER database site. When enabled, this parameter sets the default date format to a MISER date and additional parameters that are required for a MISER site (if they aren't already set): **Flatten all OCCURS**; **Automated virtual data sets**; **Allow Null dates**.

Preserve DMSII MASKING option (auto_mask_columns)

Makes the SQL Server client use the default data masking option for all columns whose corresponding DMSII item is masked in DMSII.

AA Values and RSNs

Use these options to specify how to represent absolute address (AA) values and (record serial numbers) RSNs. RSNs are unique serial numbers that get assigned to records when they get created and remain associated with the record for the life of the record. (You must explicitly enable RSNs in the DASDL.)

Use binary AA Values (use_binary_aa)

Represents AA values (including RSNs, visible RSNs, Parent_AA values and DMSII Links) as binary data, which reduces the storage requirement by half. Instead of using 12 bytes, which is typically required by character data, these values are stored using 6 bytes: binary(6) for SQL Server or raw(6) for Oracle.

Use character AA Values

Represents AA Values, Parent_AA values, RSNs, visible RSNs and DMSII LINKS as CHAR(12), where each character is the hexadecimal representation of the corresponding digit (half-byte) in the A-Series word.

Use decimal AA Values (use_decimal_aa)

Represents AA values, Parent_AA values, RSNs, visible RSNs, and DMSII LINKS as a numeric data type instead of char(12). The data type varies from database to database. For SQL Server, depending on the setting of the use_bigint parameter, either BIGINT or DECIMAL(15) is used. For Oracle, NUMBER(15) is used.

Advanced Parameters (Customizing)

Use the following parameters to customize the selected data source before you clone it. Configuration file parameters are included in parentheses.

Global data set options

Clear duplicate records encountered during data extraction (clr_dup_extr_recs)

Removes false duplicate records caused by long cloning of an active DMSII database, allowing the index creation and fixup phases to continue. If false duplicate records aren't manually removed, index creation fails. This parameter globally sets the ds_options bit that corresponds to the data set property **Clear Duplicate Extract Records** for all DATASETS table entries.

Force AA Values as indexes (force_aa_value_only)

Do not force let's the Databridge Engine decide when to use AA values. **Always force** forces the `define` and `redefine` commands to use AA values as the index, even if the data set has a SET that qualifies for use as an index. **Force only if RSN** forces the `define` and `redefine` commands to use RSN values (when they exist) as the index. This parameter globally sets the ds_options bit that corresponds to the data set property **Use AA Values (or RSNs) as Keys** for the DATASETS table entries.

Ignore new columns (suppress_new_columns)

When this parameter is set to true, the `redefine` command sets the active column to 0 for new DATAITEMS and DATATABLES entries associated with the reorganization. If you later decide that you want to include such columns, you must disable this parameter and run a `Redefine` command by clicking **Data Source > Advanced > Redefine (with options)** and checking the Redefine All Data Sets option, unless you are using the Client configurator.

Update changed columns only (minimize_col_updates)

The checkbox **Update changed column only** specifies whether the `define` or `redefine` command should set the DSOPT_Optimize_4_CDC bit (value 0x80000) in the ds_options field of the DATASETS table. This bit indicates that client will only update columns whose values have changed. To do this, stored on procedures are abandoned in favor of pure SQL without the use of host

variables. This usually slows the update speed of the Client, but when using SQL Server or Oracle replication, the overall process ultimately takes less time because significantly less data is sent to the remote database during the replication.

CAUTION: Using this parameter will significantly slow update processing by the Client. If you are replicating your relational database, enabling this feature may provide some benefit if replication is very slow.

Optimize SQL updates (optimize_updates)

Eliminates all redundant updates. Use this option when the number of occurrences for items is very large and you are not flattening the OCCURS clauses. An OCCURS clause is a DMSII construct that describes the number of times an item is present in a data set. This parameter globally sets the ds_options bit that corresponds to the data set property **Optimize SQL Updates** for all DATASETS table entries.

Split variable format data sets (split_varfmt_dataset)

This option provides an alternate way for mapping variable format data sets to tables in the relational database. The fixed parts of all variable format records are stored in the table for type 0 records. The variable parts of the records and keys are stored in tables for other types of records.

The client normally stores each record type in a table whose name is the data set name (in lower with all dashed changed to underscores) followed by "_typennn" (where *nnn* is the variable format record type). The table for type 0 records, which have no variable part, does not have a suffix like the tables for all other record types.

This parameter globally sets the ds_options bit that corresponds to the data set property **Split variable format data sets** for all DATASETS table entries.

When this option is disabled, each individual record type in the variable data set is mapped to a separate secondary table.

Use stored procedures in updates (use_stored_procs)

This parameter makes the `process` and `clone` commands generate the actual SQL command instead of a stored procedure call to perform an update. The Client still uses host variables, as was the case with stored procedures calls. Executing the SQL directly eliminates some overhead and makes processing the update faster. On exit from the Client Configurator, the Client Console command will ask you to run a `reorg` command, which creates a new set of scripts for creating the tables. It also will refresh the stored procedures for all data sets by dropping them if they exist and are no longer needed, or by recreating them if they are needed.

Table layout

Flatten all OCCURS (flatten_all_occurs)

Creates a new column in the primary table for each OCCURS item. Enable this parameter if the DMSII data contains a lot of OCCURS clauses that you want to flatten. When this parameter is disabled, a DMSII data set that has an OCCURS clause in an item will be placed in a secondary table in the relational database. As a result, a single DMSII update can end up updating multiple tables multiple times, as each occurrence of the item is placed in a separate row in the secondary table.

Maximum columns in tables (maximum_columns)

Use this parameter to limit the maximum number of columns that can be created in split tables. Split tables are created from a data set that exceeds the maximum number of columns allowed by the relational database. For SQL Server, this number is 1024. If you set this parameter to 1000, the split will occur after 1000 columns.

Indexes

Use clustered indexes (use_clustered_index)

(SQL Server only) Enable to use clustered indexes for all tables. To override this option for a single table, disable the **Use Clustered Index** check box in the **Relational Properties** pane for the table in Client Configurator.

Use Primary Keys (use_primary_key)

Creates a primary key instead of using a unique index for all tables. To override this option for a single table, disable the **Use Primary Keys** check box in the **Relational Properties** pane for the table in Client Configurator.

Table reorganization options

Use internal clone for reorganizations (use_internal_clone)

This parameter affects the `redefine` and `reorg` commands and the Client Configurator. Instead of using ALTER commands to add, delete or modify new columns to tables, the Client uses a combination of scripts and table renaming commands to create new copies of the tables with the new layouts. The Client copies the data using SELECT INTO in the case of SQL Server and CTAS (Create Table As Select) in the case of Oracle. This operation works like the bulk loader and is sometimes faster than using ALTER and UPDATE commands, but more importantly, the command is not logged. The only drawback of this method is that it requires sufficient free disk storage to hold a second copy of the table for the duration of the operation.

Reorg command update batch size (reorg_batch_size)

This parameter determines the size of the transactions that the Client uses during a `reorg` command to set the value of newly-added columns to their initial values, as defined in the DASDL. The Client Configurator creates a `reorg` script that uses a stored procedure to do the updates in batches that are executed as transactions. For a large table, this process can take quite long, but it does not run the database out of log space. Consider using the internal clone option instead.

Do not set initial values for new columns (inhibit_init_values)

Inhibits the `reorg` command from setting new columns to their INITILAValue in DMSII.

History Tables

Use the following parameters to customize the selected data source before you clone it. Configuration file parameters are included in parentheses.

Data set history tables

These parameters define the initial value of two data set property bits, which we show as **History Tables (None, Save History, History Only)**. You can override a setting for individual data sets by changing the setting in the data set properties. The `history_tables` parameter applies to all three options.

None

No history tables are created.

Save history

Creates normal and history tables in the DATATABLES client control table. The normal table replicates the host data set. The history table contains a record of the updates, including the update type, the data for the update record, and the sequence. A data warehouse can receive the synchronized clone from the normal table first. History tables use the original table name with "_h" appended to it. After periodically importing the history table into the data warehouse, the history table can be purged.

Save history only

Creates only history tables. These tables receive cloned data and updates, in addition to records of the updates. The normal data table is not created or populated.

Options

Enable dynamic history (enable_dynamic_hist)

Lets you add history tables without recloning all of the affected data sets. You can specify the default history columns in this dialog box (**Customizing > User Columns**). When you're done, you'll need to run the `Reorganize` command to create the history tables and their indexes.

Inhibit Drop (inhibit_drop_history)

Prevents the Databridge Client from inadvertently dropping history tables during a clone, process, or drop command.

This is a safeguard to prevent an unrecoverable error. If you drop tables, it's recommended that you reenable this option when you restart the client.

SQL Data Types

Use the following parameters to customize the selected data source before you clone it. Configuration file parameters are included in parentheses.

Default SQL data types

Use bigint for integers greater than 32-bits (use_bigint)

(SQL Server only) Use this option for DMSII data that is too large to fit in the int data type (32-bit integer), to bigint (64-bit integer). If the DMSII number has more than 18-digits, a data type of DEC(N) is used, as the value is too large for a 64-bit integer.

Use date data type (use_date)

(SQL Server only) Use this parameter to make the Client Configurator and the `define/redefine` commands interpret the `DIOPT_Clone_as_Date` bit in the `di_options` column of the `DMS_ITEMS` table as a request to use a data type of date instead of `smalldatetime`. This eliminates the need to set the `di_options` bit `DIOPT_Use_Date` for every item to be mapped to a data type of date.

Use datetime2 data type (use_datetime2)

(SQL Server only) Use this parameter to make the `define/redefine` commands interpret the `DIOPT_Use_LongDate` bit in the `di_options` column of the `DMS_ITEMS` table as a request to use a data type of `datetime2` instead of `datetime`. This eliminates the need to set the `di_options` bit `DIOPT_Use_LongDate2` bit for every item to be mapped to a data type of `datetime2`.

Use time data type (use_time)

(SQL Server only) Use this parameter to make the Client Configurator and `define/redefine` commands interpret the `DIOPT_Clone_as_Time` bit in the `di_options` column of the `DMS_ITEMS` table as a request to use a data type of time instead of a numeric time. This eliminates the need to set the `di_options` bit `DIOPT_Use_Time` for every item to be mapped to a data type of time..

Use varchar (use_varchar), minimum length (min_varchar)

Enable **Use varchar** to map DMSII ALPHA data to `varchar` (Microsoft SQL Server) or `varchar2` (Oracle) instead of `char`.

When you enable **Use varchar**, you can also specify a **minimum length**. Type a value (greater than zero) to map items whose length is less than the specified value to the `char` data type instead.

Use clob data type (use_clob)

(Oracle only) Maps DMSII ALPHA data that is too large to fit in a `varchar2` column to a data type of `CLOB` (instead of truncating the data or splitting it into two columns). The `varchar2` column is limited to 4000 characters.

SQL Suffixes

Use these parameters to create and assign suffixes to SQL statements that create indexes and tables. (These SQL statements are created by the `generate` command.) By adding suffixes, you can add more SQL command specifications to the end of these SQL statements. You can reference these suffixes in the **Relational Properties** pane of the Client Configurator.

Suffixes can be up to 256 characters in length. Configuration file parameters are included in parentheses.

Create table SQL statement suffix

Default suffix (global_table_suffix)

Allows you to specify a tablespace or other SQL command specification to all of the `create table` SQL statements that the Client generates, except for statements associated with a specific suffix.

Suffix (create_table_suffix[n])

Specify extra attributes for the `create table` SQL statements that the Client generates for any given table.

From the menu, assign a number so that you can reference it. Individual tables can select one of the suffixes by specifying this value in the `create_suffix` column of the corresponding `DATATABLES` client control table entry. The table suffix is then concatenated to all `create table` SQL statements for tables that specify this number.

Create index SQL statement suffix

Default suffix (global_index_suffix)

Allows you to add a tablespace or SQL command specification to all `create index` SQL statements that the Client generates, except for statements associated with a specific suffix. When you use a global suffix for an index, the Index value is zero (0) in the **Properties** pane.

Suffix (create_index_suffix[n])

Specify extra attributes for the `create index` SQL statements that the Client generates for any given table.

From the menu, assign a number so that you can reference the attribute. Individual indexes can select your defined attributes by specifying this number in the `index_suffix` column of the corresponding DATABLES client control table entry. The index suffix is then concatenated to all `create index` SQL statements for this table.

Translations

Use the following parameters to specify character translation settings. Configuration file parameters are included in parentheses.

NOTE: If you customize the character translation tables when you populate the data tables the first time, you must use them on all subsequent updates or the data will be invalid.

Use external translation library (eatran_dll_name)

Enables translation of 16-bit character sets via an alternate data translation routine, which is useful if you have multiple databases that use different languages. You can specify a translation library for the data source or use the default. Windows clients use an external DLL (the default is `DBEATRAN.DLL`). UNIX clients use a shared library (the default is `DBEATRAN.SO`).

Use configured translations

Use this option to redefine how a character gets translated. From the menu, specify the numeric value that represents the EBCDIC character code and enter the corresponding ASCII character code. You can use decimal or hexadecimal (for example, 124 for decimal or 7C for hexadecimal) to represent the character codes.

You cannot specify characters that are constant across national characters. These characters include the space, hyphen (-), single quotation mark ('), digits 0-9, and letters of the alphabet (A-Z and a-z).

User Columns

User columns let you add non-DMSII information to the relational database tables. For example, you can add an **Audit Timestamp** column to store the audit file timestamp and track when the data was last updated. (The configuration parameter for user columns is `external_column[i] = "name", sql_type, sql_length.`)

The following is a list of the different types of user columns that are available. Options for each column (such as **Add**, **Secondary Tables**, **History Tables**) are described in detail below.

Non-DMSII Column

Audit Block Serial Number

A column that contains the Audit Block Serial Number (ABSN) of the block in the audit trail from which updates are currently being processed. If you use a decimal number its precision must be at least 10.

Audit File Number

A column that contains the Audit File Number (AFN). If you use a decimal number, its precision must be at least 4; otherwise, the value may be too large and result in a SQL error.

Audit Timestamp

A column that contains the audit file timestamp of the block from which updates are currently being processed. This value is stored using a date/time data type. For extract records this column is NULL.

Audit & Extract Timestamp

A column that contains the audit file timestamp of the block from which updates are currently being processed. This value is stored using a date/time data type. For extract records this column takes on the date/time value of when the data extraction started, instead of being NULL.

Create Time

Time the record was created in the relational database (relational database time).

Data Source ID

A column that contains the data source identifier, as defined in the `data_source_id` column of the DATASOURCES client control table.

Data Source ID Key

This column is identical to **Data Source ID** except that this column is used as a key.

Data Source Name

A column that contains the data source name.

Delete Sequence Number

This column augments the **Deleted Record** column with a sequence number to provide higher granularity and avoid creating duplicate deleted records.

Deleted Record

When this column is added to a table, deleted records are marked as deleted and left in the table. The client makes this column part of the index, which allows multiple instances of a deleted record to exist in the table and not be considered duplicates. The value of this column is measured in seconds.

Include the **Delete Sequence Number** column if this column does not provide enough detail to avoid duplicate records (for example, a record is deleted twice in the same second).

Identity Column

(SQL Server only) Identifies the column using the sequence number assigned to the record when the record was created. Updates have no effect on this number.

Sequence Number

A sequence number used in history tables to determine the order of updates when they have the same update_time values. For SQL Server, the **Identity Column** for history tables is preferable, as it doesn't have this problem. For optimal results, let the client choose the default column for history tables.

Server Update Time

Time the update was applied to the relational database.

SQL Server Timestamp

Use this column (or the **Identity column**) for compatibility with older SQL Server databases. SQL timestamps are binary numbers that indicate the relative sequence in which data modifications took place in a database. The timestamp data type was originally implemented to support the SQL Server recovery algorithms.

Update Type

In history tables, this column indicates the type of update involved (insert, delete, or update). For non-history tables, this column indicates the type of the last update performed. (A value of 0 is assigned to this column during data extraction.)

Update Type (Logical Delete)

A column used in tables that preserve one deleted record per index value.

User Column 1

A generic user column whose entry is NULL. To add a default value to this type of column, use the Data table creation user script. See the *Databridge Client Administrator's Guide*.

User Column 2

A generic user column whose entry is NULL. To add a default value to this type of column, use the Data table creation user script. See the *Databridge Client Administrator's Guide*.

User Column 3

A generic user column whose entry is NULL. To add a default value to this type of column, use the Data table creation user script. See the *Databridge Client Administrator's Guide*.

User Column 4

A generic user column whose entry is NULL. To add a default value to this type of column, use the Data table creation user script. See the *Databridge Client Administrator's Guide*.

User Column Options

Use the following options to include and modify user columns in your relational database tables:

Add

Adds the selected user column to your relational database.

If you change the name, SQL type, or SQL length using these settings, those changes will be in effect if you later add user columns in the data set properties via the Client Configurator.

Secondary Tables

Includes the user column in any secondary tables

History Tables

Includes the user column in history tables.

Column Name

Shows the default column name. Type in the box to rename the column.

Data Type

Allows you to specify the data type for the column. Available data types are determined by the type and version of the relational database.

Data Length

Allows you to specify the length for the data type if a length is required. Most user columns impose a minimum and maximum length for the data type. If the value you enter does not fall into the allowable range, you will get the error "Invalid SQL Length" and will then need to correct the length before input will be accepted.

User Scripts

Use the following options to specify locations where you can save and archive user scripts. Configuration file parameters are included in parentheses.

Check for missing user scripts (check_user_scripts)

Prompts Databridge Client to return a warning if user scripts cannot be found and executed. User scripts are executed immediately after the Databridge Client scripts that are used to create tables and indexes during `process` and `clone` commands.

User scripts directory (user_script_dir)

(Required) Specify the location to which your data source customizations are saved as user scripts. If a problem occurs, you can use these scripts to restore your client control table customizations using a `Define/Redefine` command.

User scripts archive directory (user_scr_backup)

(Optional) Specify the location to which archival copies of user scripts are saved when you run the `Create Scripts` command. If a directory is not specified, scripts are archived to subdirectories of the user script directory.

Logging/Tracing Parameters

Specify your preferences for log and trace files. Legacy configuration parameter names are included in parentheses.

Client Log

Specify your settings for the client log files. The Client creates a log file in the "logs" subdirectory of the working directory for the corresponding data source. Both client programs (DBClient or dbutility) use this log file. Its naming convention is `<prefix>YYYYMMDD[_HHMMSS].log`.

DBCIntCfgServer is a Client Console support program that creates log files in the same directory. DBCIntCfgServer logs uses the same naming convention, but includes "_cfg" after the prefix.

Whenever DBCIntCfgServer executes a `define`, `redefine`, `generate` or `reorg` command, its log output is placed in main log file instead. This ensures that all the relevant information is in one log file.

Generate single output messages in log file (`single_line_log_msgs`)

Causes the client to generate all log output as single-line messages. This parameter exists to assist some log file analysis programs that fail to parse multi-line output messages.

Start new log on new day (`newfile_on_newday`)

Use this parameter to determine if the program creates a new log file, when it's created, and if the log file was created on an earlier date. A new log file is created when the date changes.

Switch log daily (`logsw_on_newday`)

Creates a new log if the client is running when the date changes.

Switch log on size (`logsw_on_size`)

Creates a new log file when the current log file reaches the size limit specified for **Maximum file size**. If new and old log files have the same creation date, a time is included to ensure a unique filename.

Maximum file size (`max_file_size`)

Specify the size limit of log files by entering a number and a suffix of K, M and G to indicate the unit of measure (kilobyte, megabyte, or gigabyte).

Filename prefix (`file_name_prefix`)

Specify a string (up to 20 characters) to change the default prefix "db" in the log filename. This can help you manage your log files if you have more than one data source. The specified prefix will be applied to both client log files and DBCIntCfgServer log files.

Service Log

(Console only) Specify your settings for service log files. Service logs store all activity related to the service and are created in the logs subdirectory of the service's working directory.

Start new log on new day (`newfile_on_newday`)

Use this parameter to determine if the program creates a new log file, when it's created, and if the log file was created on an earlier date. A new log file is created when the date changes.

Switch log daily (`logsw_on_newday`)

Creates a new log file if the date changes when the service is running.

Switch log on size (`logsw_on_size`)

Creates a new log file when the current log file reaches the specified size limit. If new and old log files have the same creation date, a time stamp is included in the filename to ensure that it is unique.

Filename prefix (`file_name_prefix`)

Specify a string (up to 20 characters in length) to change the default prefix "cp" of the log files.

Maximum file size (`max_file_size`)

Specify the size limit of service log files by entering a number and a suffix of K, M and G to indicate the unit of measure (kilobyte, megabyte, or gigabyte).

Trace

Specify your settings for trace files. To specify activities to trace, use the **Default Trace and Log Options** dialog box (**Run > Trace and Log Options**).

Filename prefix (file_name_prefix)

Specify a string (up to 20 characters in length) to change the default prefix "trace".

Maximum file size (max_file_size)

Specify the size limit of trace files by entering a number and a suffix of K, M and G to indicate the unit of measure (kilobyte, megabyte, or gigabyte).

Processing Parameters

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

General

Automatically reclone after table purge or garbage collection reorg (auto_reclone)

When enabled, changes in DMSII record locations caused by reorganizations are handled automatically by recloning the data set as soon as AA values are discovered to be invalid.

Display active data sets only in display command (display_active_only)

Shows only active data sets when the **Display** command is executed.

Number of auxiliary (ODBC/OCI) statements (aux_stmts)

Set the number of database API (that is, ODBC or OCI) statements that can be assigned to individual SQL statements. Using multiple database API statements allows SQL statements to be parsed once and executed multiple times, provided the statement is not reassigned to hold another SQL statement. Increasing the number of database API statements significantly improves processing time, if your system has enough memory. A value of zero indicates that the Databridge Client should parse and execute all SQL statements using a single database API statement.

NOTE: Do not exceed the value of the open_cursors parameter defined in the database initialization file for the Oracle instance (initSID.ora, where SID is the name of the instance). The session uses one additional statement for all control table updates.

Rollback segment name (rollback_segment)

(Oracle only) This parameter makes the client use the specified rollback segment by executing the SQL "SET TRANSACTION USE ROLLBACK SEGMENT Rollback_segment_name" at the start of every transaction.

Audit unavailable action

Wait and retry (use_dbwait)

When this parameter is enabled, the Databridge Engine (DBEngine or Enterprise) enters a wait-and-retry loop when it reaches the end of the audit trail instead of stopping the client with an end-of-audit status. Enable this option if you use real/time replication.

Use the following two parameters to control the wait-and-retry loop:

Retry interval (in seconds) (max_retry_secs)

Specifies the time interval between retries by the Databridge Engine when it is in a wait-and-retry loop. If the Databridge Engine finds that new entries written to the audit, it will start processing where it left off. Otherwise, it will wait until the next retry time.

Maximum wait time (in seconds) (max_wait_secs)

Specifies the maximum amount of time that the Databridge Engine stays in a wait-and-retry loop before it stops the client. A value of 0 for this parameter is interpreted as a request to the Databridge Engine to wait and retry forever.

Incremental wait time (in seconds) (max_wait_secs second value)

Specifies the incremental value of **max_wait_secs** that is optionally used to break up large wait times into smaller increments by making the client repeatedly issue DBWAIT calls using this second value, which must be smaller than the first value (unless the first value is 0). For example setting **max_wait_secs** to 3600,60 will result in the client issuing a DBWAIT remote procedure call with a **max_wait_secs** value of 60 seconds. Upon getting a "no more audit available" return status, the client will issue another DBWAIT call until it has received no data for the amount of time indicated by the first parameter.

This way of doing things ensures that an idle line has some traffic on it, which makes it possible to detect situations where the network goes down and neither side knows about it. Upon receiving data the client resets the timer that keeps track of the time during which no updates are received. A value of 0 for this parameter makes the Databridge Engine handle the wait-and-retry loop without any involvement by client.

Store NULL DMSII numbers as (null_digit_value)

000... Stores NULL DMSII numbers as zeros (0).

999... Stores NULL DMSII numbers as nines (9).

Advanced Parameters (Processing)

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

General

Automated virtual data sets (automate_virtuals)

Enable this parameter to preserve the link between virtual data sets and the real data sets from which they are derived during a clone or process command. (This assumes that you have already set up links between the virtual data sets and the real data sets from which they are derived. You can use the Client Configurator to do this.) The client uses this link information to determine the processing order of data sets during data extraction and to ensure that whenever you select a virtual data set you also select its parent data set.

Enable update changed columns only (enable_minimized_col)

When the Update changed columns only parameter is enabled and the Client Configurator or the Define/Redefine command is run, the DSOPT_Optimize_4_CDC bit is set in the ds_options column of all data sets. This parameter allows you to temporarily override the setting of this bit by disabling this parameter that is normally enabled.

Enable optimized SQL updates (enable_optimized_sql)

When the Optimize SQL updates parameter is enabled and the Client Configurator or the Define/Redefine command is run, the DSOPT_Use_bi_ai bit is set in the ds_options column of all data sets. This parameter allows you to override the setting of this bit by disabling this parameter that is normally enabled.

Preserve deleted records during a reclone (preserve_deletes)

Preserves deleted records when the data set is recloned.

Multithreaded updates

Number of DMSII buffers (n_dmsii_buffers)

Specify the number of RPC buffers. Raising this value may improve performance by ensuring there are enough buffers queued to keep the update workers busy at all times.

Number of update threads (n_update_threads)

Specify the number of update threads, which are responsible for executing SQL to update the user tables. If you have disk arrays, setting this parameter to a high value will increase the update processing speed at the expense of additional memory. Avoid setting this parameter to 1, as this effectively passes off all updates to the single worker thread, when executing them directly is preferable.

SQL execution timeout values (in seconds)

Query alert threshold (sql_exec_timeout - first value)

This parameter allows you to override the default setting of 3 minutes (180 seconds), for time after which the client issues a WARNING about the query taking too long to complete.

Query abort threshold (sql_exec_timeout - second value)

This parameter allow you to define a secondary timeout value for a SQL query after which time the query is aborted. This value must be greater than the value specified for the Query alert threshold described above.

Server inactivity timeout (in minutes) (max_srv_idle_time)

This parameter allows you to define a timeout value for aborting the server connection after several inactivity WARNINGS. You should use a value large enough that you at least see one of the WARNINGS.

Date and Time Parameters

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

Date parameters

Century break (yy) (century_break)

Specify a two-digit year value (yy) for dms_subtypes in the 30s, 60s, and 90s. Values less than 50 are 21st century years (20yy) and values equal to or greater than 50 are 20th century years (19yy). For example, if a value of 50 (default) describes the range of years between 1950 and 2049. In order to specify a value you must first enable the checkbox to the left of **Century break (yy)**. If century break is disabled, the default value that is dynamically adjusted based on the current year will be used.

LINC date base year (yyyy) (linc_century_base)

Specify a four-digit year value to signify the base year for LINC database dates.

Set LINC date 0 to NULL (set_lincday0_to_null)

When enabled this parameter indicates that a date value of 0 should be treated as NULL, rather than 1/1 of the LINC date base year.

Correct invalid date values (correct_bad_days)

Correct only bad days <= 31 tells the Client to attempt to correct DMSII dates with a bad day value by setting the value to last day for the given month and year. **Correct all bad days and months** allows the Databridge Client to perform the following corrections in addition to the ones for the case where correct_bad_days is set to 1. Day values greater than 31 are set to the last legal day of the month, month values greater than 12 are set to 12, and a month value of 0 is set to 1.

When **Do not correct** is selected, the client changes a day value of 0 to 1, which makes it possible to handle partial dates like credit card expiration dates. If **Only correct days that are 0** is selected, the client treats such dates as being in error and sets them to NULL.

If the DAOPT_Nulls_Allowed bit in the da_options column of the corresponding DATAITEMS entry is not set, the bad date is stored as 1/1/1900 in SQL Server.

Default date formats

Numeric date (numeric_date_format)

Specify the date format used to encode numeric dates. Options include all six- and eight-digit numeric data formats.

Default date format (default_date_fmt)

Specify the default date format that appears when you select the **Clone as Date** option from the DMS item context menu in the **DMSII** pane of the Client Configurator.

Null date values for columns that do not allow NULL

Datetime

Changes the value used to represent a NULL date in a datetime column that does not allow nulls.

Datetime2

Changes the value used to represent a NULL date in a datetime column that does not allow nulls.

DBEngine and DBEnterprise Parameters

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

General

DBEngine (Extract) workers (engine_workers)

Overrides the DBEngine WORKERS=*n* parameter on the host which controls the number of extract workers the Databridge Engine can use during the data extraction phase.

Convert reversals to updates (convert_reversals)

The radio buttons **User server setting**, **Don't convert**, and **Convert** allow the Client to override the Databridge Engine control file parameter CONVERT REVERSALS. Refer to the Databridge Host Administrator Guide for more details on this parameter.

Enable DOC records (for debugging only) (enable_doc_records)

Requests DOC records from the Databridge Engine (DBEngine or Enterprise Server). Use only for troubleshooting the Databridge Engine problems.

Include latest StateInfo in record headers (use_latest_si)

Requests StateInfo in all data records sent by the server during audit file processing. If the DMSII audit file records have an associated timestamp, this option can result in more accurate audit timestamp values.

Verify DBEnterprise audit origin (dbe_dflt_origin)

Specifies the expected origin for Enterprise Server audit files during normal operations. Options include: **Cached**; **Indirect disk**; and **Direct disk**.

COMMIT parameters

Use these parameters to override Databridge Engine CHECKPOINT parameters.

Commit every (n) ABSN blocks (commit_absn_inc)

Specify a value to override the Databridge Engine CHECKPOINT CLIENT EVERY *nnn* SECONDS AUDIT BLOCKS parameter setting. The Databridge Engine will generate a commit at the next quiet point after the specified number of audit blocks have been processed.

Commit every (n) SECONDS (commit_time_inc)

Specify a value to override the Databridge Engine CHECKPOINT EVERY *nnn* SECONDS parameter setting. The Databridge Engine will generate a commit at the next quiet point after the specified number of seconds has passed since the last commit.

Commit every (n) TRANSACTIONS (commit_txn_inc)

Specify a value to override the Databridge Engine CLIENT EVERY *nnn* TRANSACTIONS parameter setting. The Databridge Engine will generate a commit at the next quiet point after the specified number of transactions have occurred since the last commit.

Commit every (n) UPDATES (commit_update_inc)

Specify a value to override the Databridge Engine CHECKPOINT CLIENT EVERY *nnn* UPDATE RECORDS parameter setting. The Databridge Engine will generate a commit at the next quiet point after the specified number of updates have been processed since the last commit.

Commit long transactions (commit_longtrans)

To have the client override the Databridge Engine CHECKPOINT LONG TRANSACTIONS parameter at the start of a process command, select either **Commit** (to set the parameter to True) or **Don't commit** (to set the parameter to False). If you don't want the client to override the parameter, select **Use DBEngine setting**.

DMSII Data Error Handling

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

Character data error

Control character

Change to space (convert_ctrl_char)

Changes a control character encountered in alpha data to a space.

Change to '?' and count as error (inhibit_ctrl_chars)

Treats a control character encountered in alpha data as an error and converts it to a question mark (?).

Translate if possible

Translates control characters that do not interfere with client operations. The client won't allow a control character to be translated to the character NULL (00) as this causes problems with NULL terminated strings. In addition, it won't allow the use of CF and LF characters, as they interfere with bulk-loader operations. If you use SQL Server and the TAB character is the bcp delimiter, the client won't allow you to translate a character to a TAB character.

Change 8-bit character to '?' and count as error (inhibit_8_bit_data)

Considers each 8-bit character in alpha data to be an error and converts it to a question mark (?).

Keep undigits in NUMBER items stored as character data (keep_undigits)

This options makes the client preserve the bad digit values in DMSII NUMBER items stored as CHAR or VARCHAR data.

Enable High Value Padding

This option lets you mark items as padded with high values in order to achieve left justification. It applies to ALPHA items and unsigned numeric items that are stored as ALPHA data.

Set item to NULL if errors exceed (%) (alpha_error_cutoff)

Specifies the percentage of data errors in any given alpha item that is tolerated before the field is declared bad and is treated as NULL.

General error handling

Discard records containing data errors (discard_data_errors)

Instructs the Client to write all records with data errors to the discard file `tablename.bad` and not apply them to the relational database.

Display data errors (in log file) (display_bad_data)

Causes the Databridge Client to display the raw DMSII data for an item that has a data error. Use this option for debugging if you encounter many data errors. Typically, this output is suppressed when the number of errors exceeds the limit specified by **Error display limit**.

Suppress duplicate (insert) warnings (suppress_dup_warnings)

Prevents warnings for duplicate inserts and failed updates from displaying during update processing.

Error display limits (error_display_limits)

Set the maximum number of errors per data set that is permitted on-screen (display) and saved in the log file.

Discard record thresholds

Set these values to control how the Client handles discarded records.

Maximum total discards is the total number of discards the Client will tolerate before abending. This value must be greater than the value for **Maximum per-table discards** unless the second value is zero, indicating that it's unlimited.

Maximum per-table discards is the maximum number of discards records for a table that are written to the discard file. Discards that exceed this number are ignored. If either of these values are set to zero, no limits are imposed for the corresponding actions, and the Client will behave the way it did before this parameter was implemented.

Error Recovery

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

Maximum retries

Specify the number of time the service tries to run a process command that terminates with a recoverable error exit status. If the exit status indicates that the Databridge Server is inaccessible or that the relational database is down, this parameter is ignored and the service retries with longer intervals until the interval reaches five minutes, after which it retries every 5 minutes.

Minimum wait (in seconds)

Specify the delay (in seconds) before a process command is attempted after an error.

Retry interval (in seconds)

Specify the interval (in seconds) between attempts by the service to launch the next process command when a process command terminates with a recoverable exit code.

Disable data source conditions

Specify a DBClient exit status that causes the service to disable the data source. For example, if you use `STOP AFTER TASK xxx` to detect the end of the banking day, you can make the Client disable the data source to prevent any further client activity when this run is detected in the audit trail.

The three conditions in question are **Stop task name reached**, which corresponds to the example above, **Stop time reached** which happens when you use `STOP AFTER TIME xxx`, and **Audit file number reached**, which happens when you set the Client to stop after a given audit file is processed.

Scheduling

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

Schedule

Scheduling parameters apply only to processing commands that are initiated by the service.

None

Select this option to initiate client runs manually.

Fixed delay (in seconds)

Select the check box to specify a fixed delay (in seconds) between the time one client run stops and another one begins for the same data source.

Daily

Using the clocks provided, specify one or more times at which the service starts a client run if a client run isn't already in progress.

Blackout Period

Using the two clocks provided, specify the start and end times of a fixed block of time during which the Client cannot run. This parameter is useful for operations, such as database backups, that can only take place when the Client is inactive. For example, if you want to back up the database daily between 1:00 a.m. and 2:30 a.m. daily, define a blackout period from 0:55 to 2:30. The extra 5 minutes ensures that the Client finishes any long transactions before the database backup begins.

Automatic Run Options

Process data source on service startup

Starts a client process command any time the service restarts.

Redefine data source when DMSII reorganization detected

Enable this option to continue processing without any interruptions after a DMSII reorganization, provided the reorganization did not affect the table layout. When the exit code for a `process` command indicates that a DMSII reorganization has been encountered, the Client starts `DBClntCfgServer` and then runs a `redefine` command to determine if the DMSII

reorganization affected the table layout. If the table layout isn't affected, the `redefine` command returns an exit status of 0, and the service immediately launches a `process` command.

If the `redefine` command determines that new scripts must be generated, manual intervention is required. To have the Client automatically generate scripts in these cases, select **Generate scripts after an automatic redefine** in addition to this option.

Generate scripts after an automatic redefine

Enable this option to have `DBCIntCfgServer` run a `generate` command when the exit code for a `redefine` or `process` command specifies that a `generate` command is required. This command creates a new set of scripts for the tables of data sets that require them. This usually indicates that the data set has been added or that the reorganization caused the data set to be recloned.

NOTE: If the exit status for the `redefine` command indicates that a `generate` command is required, this means that one or more data sets will be recloned. If you do not want clone commands automatically started, refrain from setting this option.

Statistics Parameters

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

Logging options

Show performance statistics (`show_perf_stats`)

Prints performance statistics at the end of the data extraction phase, after the processing of an audit file finishes (i.e. when audit file switch occurs) and at the end of the run.

Show statistics (`show_stats`)

Displays record count statistics at intervals that you specify using the parameter **Record count display intervals**. This can be useful for monitoring the progress of lengthy operations.

Show table statistics (`show_table_stats`)

Controls whether or not the table statistics are displayed after an audit file switch. When this parameter is enabled, the number of updates performed on each table are displayed, in addition to the performance statistics.

Record count display intervals

Data extraction (statistics_increment - part 1)

Specify the frequency at which data extraction statistics are displayed.

Update processing (statistics_increment - part 2)

Specify a frequency at which update processing statistics are displayed.

Stop Conditions

Specify processing settings for the selected data source. Configuration file parameters are provided in parentheses.

Stop after data extraction phase (defer_fixup_phase)

Defers the fixup phase to the next **Process** command.

Stop after fixup phase (stop_after_fixups)

Stops the Client when the fixup phase is complete. At that point, all data sets will have a ds_mode value of 2 and all of the tables in the relational database will be consistent.

Stop after garbage collection reorg (stop_after_gc_reorg)

Stops the program at the first quiet point after a garbage collection reorganization.

Stop after Enterprise Server mode change (stop_on_dbe_mode_chg)

Enabling this parameter causes the program to stop as soon as it detects that the “Databridge Enterprise Server access mode” changes from the value specified in the parameter **Verify DBEnterprise audit origin**. If this parameter is set to “direct” and DBEnterprise switches to “indirect”, this will result in the Client stopping at the next quiet point.

Dynamic stop conditions

Stop task name (stop_condition_task)

Causes the Databridge Engine to stop processing when it encounters the specified task. This task must be a DMSII program (that is, a program that opens the DMSII database). The **After** or **Before** options indicate whether processing stops at the first quiet point *after* or *before* the task is encountered in the audit trail. Consult the DMSII personnel to determine the syntax of the task name.

Stop time (stop_condition_time)

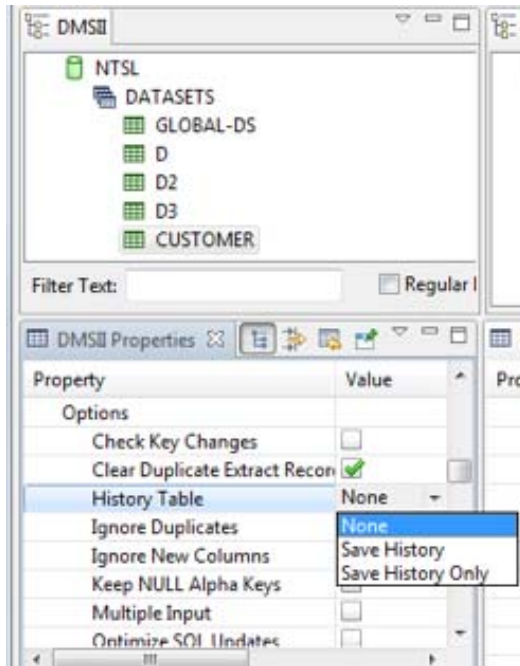
Causes the Databridge Engine to stop processing when it encounters an audit block whose timestamp value is greater than or equal to the specified time. The Client automatically provides the date part of the time stamp based on the current audit location being processed. The **After** or **Before** options indicate whether processing should stop at the first quiet point after or before the audit block whose timestamp value causes the Databridge Engine to stop processing.

4 Customize a Data Source

NOTE: Use the Client Configurator to map individual DMSII data sets (and their items) and customize the table layout of the relational database. Customizations should be made before you clone a data source or your changes may require that you reclone. For a description of the Client Configurator graphical user interface, see [“Tour the Client Configurator” on page 9](#). If you want to make customizations that affect the entire data source, use the **Client Configuration** dialog box. See [“Customize the Client Configuration” on page 23](#).

To customize the table layout

- 1 To start the Client Configurator, from the **Explorer** pane, right-click a data source and click **Customize Data Source** in the context menu.
- 2 In the Client Configurator, in the **DMSII** pane, select the data set you want to customize. In the example below, EMPLOYEE is selected.





Properties for the selected data set appear in the **DMSII Properties** pane.

For best results, make all of your changes in the **DMSII Properties** pane first, and then use the **Relational Properties** pane, rather than alternating between the two panes.

- 3 To customize the data set, do one or more of the following:
 - ♦ To exclude a data set from replication, uncheck the **Active** check box.

- ◆ To modify the data set, make selections from the **DMSII Properties** pane.
 - ◆ To modify a DMS item, right-click it in the **DMSII** pane and choose from the available options in the context menu that reflects the options available for the given item. (See individual topics in this section.)
- 4 To customize the relational database tables and their columns, do one or more of the following:
 - ◆ Rename tables or columns. See [“Rename a Table or Column” on page 60](#).
 - ◆ Rename indexes for tables.
 - ◆ Change the SQL type and the precision and scale of column. See [“Change the SQL Data Type” on page 61](#).

CAUTION: Changing the SQL type of the column can create a table with a record size that is too big for the relational database.

- 5 To undo a change, right-click the modified item and click **Discard Changes** in the context menu.
- 6 To commit a change, in the toolbar, click **Save** . If the **Save** button is unavailable, click anywhere in the Client Configurator window to activate it.
- 7 In the toolbar, click **Refresh Relational**  to populate the **Relational** pane. This pane shows how the modified items will be mapped to the relational database.
- 8 Choose **File > Exit** from the **Menu bar** to close the Client Configurator.

DMSII vs. Relational Database Terminology

Before you begin the data mapping process, it may be helpful to review the terminology that each environment uses to describe structures.

DMSII Term	Relational Database Term
Data set	Table
DMS item (data element, field)	Column (data item)
Key	Key
Record	Row (record)
Set	Index

NOTE: A relational database index is a set of column names that is used to efficiently access a row (of a table).

Search for DMS Items

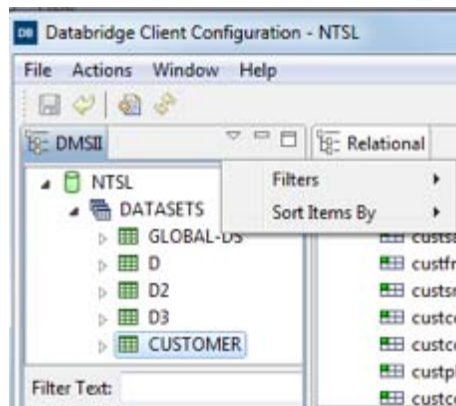
Use the **Filter Text** box in the Client Configurator to locate DMSII items that you want to view or modify. You can use text or regular expressions (regex) to filter out items of interest.

To search for items

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source**. The Client Configurator opens.
- 2 At the bottom of the **DMSII** pane, click in the **Filter Text** box.

Filter Text:

- 3 Do one of the following:
 - ♦ To find items that contain specific text in their name, type that text in the box. For example, type "DT" (without quotation marks) to show only columns that contain the string "DT" anywhere in their names.
 - ♦ To find items that match a particular pattern, type a regex expression and click the **Regular Expressions** check box.
- 4 Click **Apply**.
- 5 (Optional) To sort the DMS items alphabetically, click the arrow in the upper-right corner of the **DMSII** pane, and select **Sort Items By > Name** as shown:




You can also filter DMS items based on whether they're active (selected for replication). The default sort method is by item number.

Cloning a Number as Alpha

You can use the Client Configurator to make the Client treat a numeric item as if it was an ALPHA item.


To clone a NUMBER as ALPHA

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the numeric item you want to clone as ALPHA and click **Clone As Alpha** in the context menu.
- 3 In the toolbar, click **Save** .

Cloning an Alpha item as a Number

You can use the Client Configurator to make the Client treat an ALPHA item that contains numeric data as if it was a NUMBER.

To clone an ALPHA item as a NUMBER


- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the alpha item you want to clone as a NUMBER and click **Clone As Number** in the context menu.
- 3 The **DMSII properties** pane allows has two entries in the Options section labeled **DMS Scale** and **DMS_Signed** that allow you to specify how the data is to be interpreted. The client automatically recognizes COBOL display data and properly converts it to a number.
- 4 In the toolbar, click **Save** .

NOTE: You need to make sure that the items contain numeric data before doing this, as you will otherwise get data errors.

Cloning an Alpha item as Binary

You can use the Client Configurator to make the Client clone an ALPHA item as a binary type (BINARY for SQL Server and RAW in the case of Oracle).

To clone an ALPHA item as binary

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the alpha item you want to clone as binary and click **Clone As Binary** in the context menu.
- 3 In the toolbar, click **Save** .

Combine DMS Items

You can use the Client Configurator to combine DMS items into a single column in the relational database by merging or by concatenating them. If the items exist side-by-side in the relational database, use the **Merge** option. Use the **Concatenate** option for non-adjacent items if both items are of the same type (numeric or alpha).

Under certain conditions, you can also combine all the items in a DMSII GROUP (including those in a GROUP within the GROUP) to form a single item when they all have the same data types. We refer to this action as collapsing a GROUP. This is only possible if the items, excluding nested GROUP items, are all unsigned numbers or all alpha items. The length of the resulting item will be that of the outer GROUP which spans all the items. For example, if you have a GROUP with two NUMBER(2) items and a NUMBER(4) item, the collapsed GROUP will be treated as if it was a NUMBER(8)


CAUTION: Make this type of change only before cloning the data source. Otherwise, you may need to reclone.

To concatenate items

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 Decide the order in which you want the concatenated items to appear.
- 3 In the **DMSII** pane, right-click the DMS item you want to appear first in the column, and click **Concatenate Item** from the context menu.

The **Concatenate With Item** property appears in the **DMSII Properties** pane.

- 4 In the **DMSII Properties** pane, click the **Values** column next to **Concatenate With Item**. From the list, select one of the available DMS items, which appear in the list box in alphabetical order.
- 5 To undo your changes, right-click the DMS item and select **Discard Changes**, or select **Concatenate Item** a second time.

- 6 In the toolbar, click **Save** .


- 7 In the toolbar, click **Refresh Relational**  to update the **Relational** pane.

When concatenating items that have been marked to be cloned as ALPHA or as NUMBER, they will become eligible for concatenation with items of the resulting type. Thus you can concatenate an ALPHA item with a numeric item that is being cloned as ALPHA and vice versa. In other words, you can perform more than one customization on the same item, as long as these operations are compatible with each other.

After you customize a data source, the Client Console will notify you that you must generate scripts. See [Define/Redefine and Generate Scripts \(page 21\)](#).


To merge items

CAUTION: Merging items can cause problems if a DMSII reorganization relocates one of the merged items. When possible, concatenate items instead. This option is included primarily for compatibility with older systems and as a method for combining three items (when used in conjunction with concatenation).

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the DMS item that you want to merge with the item that immediately follows it and click **Merge Item** from the context menu.
- 3 In the toolbar, click **Save** .

NOTE: You can concatenate a merged item with another item, excluding another merged item and vice versa.



To collapse a group

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the GROUP you want to collapse and click **Collapse Group** in the context menu. The menu will only have this option if the GROUP qualifies for being collapsed.
- 3 In the toolbar, click **Save** .

Split a DMS Item

CAUTION: Make this type of change only before cloning the data source. Otherwise, you may need to reclone.


To split a single item into two columns

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the DMS item and click **Split Item** from the context menu.
- 3 In the properties pane for the item, set the value of the Offset For Split (this is initially set to 2)
- 4 In the toolbar, click **Save**  and then click **Refresh Relational** .
In the **Relational** pane, two new columns are added (*column_x1* and *column_x2*) to replace the original column. You can then rename these two columns.

Cloning an Item as a Date Type

You can use the Client Configurator to clone DMS items as relational database date data types.


To clone an item as date

- 1 In the **Explorer** view, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the alpha item you want to clone as a date and click **Clone As Date** in the context menu.
- 3 Examine the **Date Data Type** and **Date Format** entries in the **Options** group in the **DMSII Properties** pane. If necessary, modify their value by expanding the list box in their associated value column and picking the desired type.
- 4 In the toolbar, click **Save** .

Cloning an Item as a Date/Time Type

You can use the Client Configurator to clone DMS items as relational database date/time data types. The client configurator distinguishes between date and date/time data types, even though in some cases these two types are the same. Oracle supports a single date type, which is actually a date/time type, while SQL Server has a variety of date/time data types and a date data type that has no associated time.

To clone an item as a date/time

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the alpha item you want to clone as a date/time and click **Clone As Date And Time** in the context menu.
- 3 Examine the **Date Data Type** and **Date Format** entries in the **Options** group in the **DMSII Properties** pane. If necessary, modify their value by expanding the list box in their associated value column and picking the desired type.
- 4 In the toolbar, click **Save** .

NOTE: You can clone concatenated items, merge items and collapsed groups as date/time provided that the resulting length is appropriate for a date/time value.

Custom Format Dates

The Client can handle variety of date and date/time encoding schemes, these formats are referred to as unique format in the *Client Administrator's Guide*. The Client handles an arbitrary string of numeric values by processing a series of 4-bit codes that describe the various fields in the custom date or date/time value. For example, this format allows the Client to deal with dates that just have a year and month by simply setting the missing day to 1.


The encoding values used are:

- 1- 4-digit year
- 2- 2-digit year
- 3- 2-digit month
- 4- 2-digit day
- 5- 2-digit hours
- 6- 2-digit minutes
- 7- 7 digit seconds

The format is represented by a hexadecimal number that looks string of decimal numbers. For example, 0x14 is a date of the form "yyyymm". The numbers are right-justified.

To clone an item as a date/time using custom format

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.


- 2 In the **DMSII** pane, right-click the alpha item you want to clone as a date/time and click **Clone As Date (Unique Format)** in the context menu.
- 3 Enter the Date Format encoding value in the Value column of the **Date Format (Unique)** entry in the **Options** group in the **DMSII Properties** pane.
- 4 Examine the **Date Data Type** entry in the **Options** group in the **DMSII Properties** pane. If necessary, modify its value by expanding the list box in its associated value column and picking the desired type.
- 5 In the toolbar, click **Save** .

NOTE: You can clone concatenated items, merge items and collapsed groups as date/time provided that the resulting length is appropriate for a date/time value.

Cloning an Alpha(36) item as a Uniqueidentifier

You can use the Client Configurator to make the Client clone an ALPHA(36) item that contains a GUID as a SQL Server UNIQUEIDENTIFIER data type.


To clone an ALPHA(36) as a Uniqueidentifier

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the alpha item you want to clone as a Uniqueidentifier and click **Clone As Uniqueidentifier** in the context menu.
- 3 In the toolbar, click **Save** .

Customization Options for REAL Item

Several customization options exist for DMSII REAL items, which include cloning the entry as a date/time, a time, and an RSN.

To customize a REAL item

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the REAL item you want to customize and click the appropriate option in the context menu. The options include **Clone As Date**, **Clone as Numeric Date**, **Clone As RSN**, **Clone As Date And Time** and **Clone As Time**.
- 3 Examine the entries in the **Options** group. If necessary, modify the **Data Type** and **Format** entries in the **DMSII Properties** pane by clicking the option's entry in the value column, clicking the arrow to open the drop-down menu, and clicking the desired type.
- 4 In the toolbar, click **Save** .

NOTE: These options are automatically set for visible RSNs, so you should not need to do any additional customization on them. Their data types will be the same as regular RSNs because the options defined in the **AA Value and RSN** section of the Client Configuration dialog also apply to visible RSNs.

Split Variable Format Data Sets

Use this option as an alternate method of mapping variable format data sets to tables in the relational database. The fixed parts of variable format records are stored in the table for type 0 records. The variable parts of the records and keys are stored in tables for other types of records.

NOTE: This type of change should only be made before you clone the data source or after variable data sets have been added because of a DMSII reorganization.

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the data source and click **Client Configuration** from the context menu.
- 3 In the **Client Configuration** dialog box, on the left, click the triangle next to **Customizing** to expand the options.
- 4 Click **Advanced Parameters**.
- 5 Click to enable **Split variable format data sets** and click **OK**.

Add a User Column

Use this procedure to add columns to tables in the relational database that contain additional information. User columns are added to the end of a table, after the DMSII data.

To add a user column to the entire data source, use the **Client Configuration** dialog box. To add user columns to specific tables, use the Client Configurator. We recommend that you add user columns before you clone your data source.


To add global user columns

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, right-click the data source and click **Client Configuration** from the context menu.
- 3 In the **Client Configuration** dialog box, click the triangle next to **Customizing** to expand the options and then click **User Columns**.


NOTE: For a full description of the available options, in the lower-left corner of the **Client Configuration** dialog box, click the **Help (?)** button.

- 4 Click the check box in the Add column for user column/s you wish to add, and then click **OK**.

To add user columns to a specific data set

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, click the triangle next to the data source to expand it and select a data set.
- 3 In the **DMSII Properties** pane, under **User Columns**, select one or more of the available types of user columns.
- 4 In the toolbar, click **Save** .

To remove a previously-added user column

- 1 Follow steps 1 and 2 above.
- 2 In the **DMSII Properties** pane, under **User Columns**, uncheck the column you wish to remove.
- 3 In the toolbar, click **Save** .

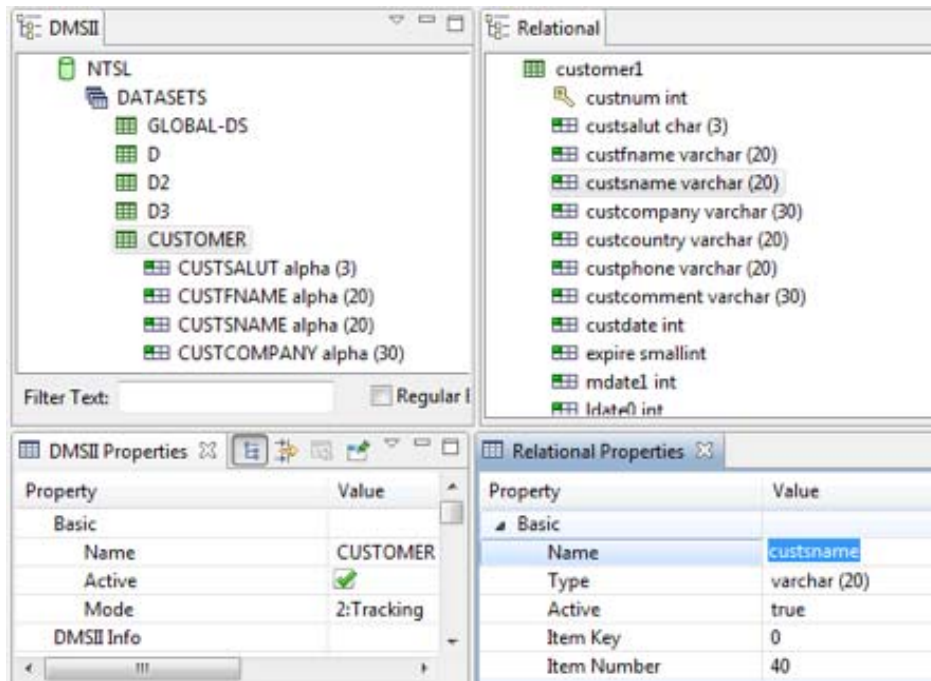
Rename a Table or Column


Use this procedure to rename a table or a column in the relational database.

CAUTION: Make this type of change only before cloning the data source. Otherwise, you may need to reclone.

To rename a table or column

- 1 In the **Explorer** pane, right-click a data source and choose **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, select the data set or data item you want to rename.
The table and columns for the selected data set appear in the **Relational** pane.
- 3 In the **Relational** pane, select the table or column you want to rename.
- 4 In the **Relational Properties** pane, do one of the following:
 - ♦ To rename a table, click the value for **Table Name**, and type the new name.
 - ♦ To rename a column, click the value for **Name**, and type the new name.




- 5 Press the **Enter** key to apply the value.
The icon for the renamed item changes to indicate that the item has one or more pending changes.
- 6 To undo your changes, right-click the modified item, and select **Discard Changes**.
- 7 In the toolbar, click **Save** .


Change the SQL Data Type

Use the following procedure to change the SQL data type in the relational database. You can change the data type by selecting a mapping option for the DMS item. If your database is already cloned, making this type of change to the relational database requires recloning unless your relational database supports data type transformations.

While it's not recommended, you can directly change the properties of the relational database columns. For example, you can changing "varchar" to "var". If you modify the length of a column, make sure that the record does not exceed its maximum size.

- 1 In the **Explorer** pane, right-click a data source and choose **Customize Data Source**. The Client Configurator opens.
- 2 In the **DMSII** pane, click the data set that contains data items you want to change.
- 3 Do one of the following:
 - ♦ Right-click a data item and click a data type (for example, **Clone as Binary** or **Clone as Date**). From the toolbar, click **Refresh Relational**  to show the change in the **Relational** pane.
 - ♦ In the **Relational** pane, select a data item. Then, in the **Relational Properties** pane, click the value for **SQL Type** and select an option from the drop-down menu.
 - ♦ Modify the length and scale values, as needed.

NOTE: If you change the data item from an int to a dec (or vice versa), you may have to save your changes before you can modify the length.

- 4 Click anywhere outside the **Value** field.
- 5 To undo your changes, right-click the modified item, and select **Discard Changes**.
- 6 In the toolbar, click **Save** .

Create Composite Keys


Use the Client Configurator to create composite keys if unique keys haven't been specified in the DMSII data. A composite key is created from a set of column names in a table that's derived from a data set.

If unique keys are not defined and you use Unisys Enterprise Database Server for ClearPath Extended Edition (formerly known as DMSII Extended Edition), the Databridge Client creates unique keys using RSNs, because they provide the best indexes. If RSNs aren't available, valid AA values are used for keys. This means that if a DMSII reorganization occurs, you will need to reclone all of your data sets. In data sets that don't have fixed length records, the AA values continually change and cannot be used as keys.

CAUTION: Before you create keys, make sure that your key (the combination of columns) will be unique and will not encounter duplicates. Otherwise, your data can become corrupted. You can only create composite keys for uncloned data sources.

Create a unique composite key

- 1 In the **Explorer** pane, right-click a data source and click **Customize Data Source** from the context menu. The Client Configurator opens.
- 2 In the **DMSII** pane, look for the items you want to include in the composite key. Click on one of them.
- 3 In the **Relational Properties** pane, find the **Item Key** property. In the value column, give it a number. For example, you may assign values of 1, 2, 3, and 4 to four columns to create your composite key. The numbers determine the order of the columns in the index.

NOTE: If you can't edit the **Item Key** value, a key may already be defined for the table. Keys appear in the **Relational** pane with this  icon.

- 4 In the **DMSII** pane, select the next item you wish to include in the composite key, go to the Relational Properties pane and set the **Item Key** to the next number in the sequence, and so on until each item is assigned an Item Key number.
- 5 When you're done, in the menu bar, click **Save**.

5 Working with Data Sources

This section helps you set up your data sources and prepare for the task of cloning.

Best Practices

For the best possible outcome from the Databridge Client software, follow these guidelines:

- ◆ Disable cloning for any non-essential DMSII data sets in the Client Configurator.
- ◆ If you clone the base structure, you should not clone the remaps. Doing so replicates the same data twice. However, if you have sensitive data in the base structure that you do not wish to replicate, it is perfectly acceptable to clone a remap instead of the base structure. To do this, modify the active column in the DATASETS tables to enable the remap and disable the base structure.
- ◆ Before you customize a data source in the Client Configurator, change its global settings in the **Client Configuration** dialog box. Global settings affect which options are available in the Client Configurator. If you change the global settings after you start the customization process, you'll need to refresh the data sets in the Client Configurator to show the correct properties. This can be a time-consuming process for large DMSII databases.
- ◆ Customize the data source *before* you clone. First, use global customization parameters in the **Client Configuration** dialog box. Then, in the Client Configurator, modify properties for individual data sets to achieve the results you want. Many of the customization parameters are difficult to change later without recloning the data source.

Set Up Client Log Files

Use this procedure to specify when a new log file will be created for the data source. You can also manually start a new log file at any time. To set up logs for the service, see [“Set Up Service Log Files” on page 73](#).

To set up log files

- 1 In the **Explorer** pane, right-click the data source and click **Client Configuration** from the context menu. The Client Configuration dialog box opens.
- 2 In the **Client Configuration** dialog box, in the left pane, expand **Logging/Tracing**.
- 3 Below **Logging/Tracing**, select **Client Log** and select the settings you want.
- 4 Click **OK**.

To manually start a new log file

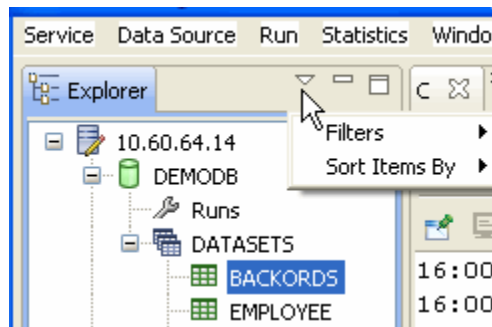
- 1 In the **Explorer** pane, select the service.
- 2 From the **Run** menu, choose **Switch Client Log**.

Show or Hide Data Sets

You can use filters to show or hide data sets based on their state. For example, if you want to limit your view to data sets that have been selected for replication, you can show *active* data sets. In the Client Configurator, you can use filters to find items that require additional attention.

To show or hide data sets

- 1 In the **Explorer** pane of the Client Console or the **DMSII** pane of the Client Configurator, select the data source or the data set that contains both active and inactive items.
- 2 In the upper-right corner of the pane, click the triangle icon to show the menu (this applies to both the Explorer and DMSII panes).




- 3 Do one or more of the following:
 - ◆ To show only active data sets, select **Filters > Active**.
 - ◆ To show nonactive data sets, select **Filters > Inactive**.
 - ◆ (**DMSII** pane only) To show data sets that have been modified and require attention, select **Filters > Items Needing Attention**.
 - ◆ To show all data sets, select **Active** and **Inactive**.

View Multiple Data Sources

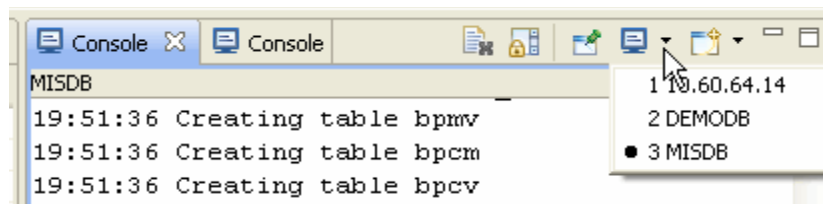
If you process multiple data sources at a time, you can switch between them from the **Console** pane, like switching between television channels. If you want to view output for each of your data sources at the same time, you can create multiple consoles and set each one to a different data source.

Having several **Console** views open at once can impact performance.

To create a new Console pane

- 1 In the far-right corner of the **Console** pane, click the menu and select **New Console View**.
By default, the new console appears as an additional tab in the current **Console** pane.
- 2 In the new console, click the **Display Selected Console** menu  and select the data source or service whose output you want to show in that pane.

NOTE: This menu is only active when a data source is processing.



Disabling, Unloading, or Removing a Data Source

This topic describes when to use the **Enable/Disable Data Source**, **Unload Data Source**, and **Remove Data Source** commands.

Disabling/enabling a data source

You can use the **Enable/Disable Data Source** command to stop all `process` commands for a data source. When a data source is disabled, you cannot run `process` or `clone` commands or update the data tables in any way. This can be useful if you run a reporting program at the end of a business day and must ensure that client runs don't interfere with your reporting software.

If an error occurs for which there is no recovery, Databridge Client may disable a data source. If this happens, you'll need to re-enable the data source using the **Enable/Disable Data Source** command.

Unloading/reloading a data source

The **Unload Data Source** command is used to back up the client control tables when they're in a good working state. When you use this command, enter a filename, and use optional command-line commands. You can use this back up copy to restore the client control tables as needed by using the **Reload Data Source** command and entering the same filename you used with the **Unload Data Source** command.

The file created by the **Unload Data Source** command is intended only for the program's use. To produce a readable version of the client control tables, see [“Display the Client Control Tables” on page 81](#).

Removing a data source

The **Remove Data Source** command deletes the data tables (that is, the cloned data) of a data source. After you select the command, you have the option to remove the working directory and all associated files for the data source, including entries in the client control tables. (If no other data sources exist, this option deletes the client control tables.)

You might want to delete just the data tables after testing configuration features on a new data source and before cloning it for real-world production.

Change the SOURCE Password

Use this procedure to regain access to the DBServer if the password to the SOURCE changes. The SOURCE password is specified by the `KEY` parameter in the DBServer control file. For more information, see the *Databridge Host Administrator's Guide*.

To change the SOURCE password

- 1 In the **Explorer** pane, right-click a data source and click **Advanced > Change SOURCE Password**.
- 2 In the **New Password** box, type the password exactly as it appears in the DBServer control file
- 3 In the **Verify Password** box, type the password again, and then click **OK**.

Change the Database Password

The database password is specified in the authentication settings when you add a new data source. For example, if you use the Windows authentication, this password might be what the database administrator (DBA) uses to log in to the relational data base.

To change the data source password

- 1 In the **Explorer** pane, right-click the data source and click **Advanced > Change Database Password**.
- 2 Type the new password in **New password** and **Verify password** boxes, and then click **OK**.

6 Security Options

This section covers security options available within the Client Console.

Manage User Permissions

Use this procedure to set user permissions for the Client Console and the Client service to a maximum of thirty user IDs.

Permissions can be granted as pre-defined roles (such as Administrator), which can be customized to include access to specific menu commands in the Client Console.

To add a user ID and assign permissions

- 1 In the **Explorer** pane, select a service and then click **Service > Manage Users**. (You can also right-click on the service to open the context menu and then click **Manage Users**.)
- 2 In the **Manage Service User IDs** dialog box, under **Actions**, click **Add User** and enter the user ID in the text box.
- 3 From the **Roles** menu, select the role that best fits the user's level of usage.

Role	Privileges
Administrator	Has full privileges to the Client and configuration, including starting client runs, and customizing data sources.
Operator	Can manually start and stop client runs.
User	Can view, but not modify, properties and operations in the Client Console.

- 4 If you want to customize the user ID's permissions, click **Apply** and continue the following steps. If you're done, click **OK**.
- 5 To customize the user ID's permissions, from the **User List**, select the user ID you just added and click **Modify User**.
- 6 Click **Customize**. In the **Customize Menu Permissions** dialog box, select the check boxes for all menu commands to which the user will have access, and then click **OK**.

The menu commands are arranged in groups that correspond to the items in the menu bar, with an additional group for **Data Source's Advanced** sub-menu.

NOTE: You cannot customize access to the **Statistics** and **Help** menu items as these items are accessible to all users.

- 7 Click **OK** to close the **Manage Service User IDs** dialog box.

TIP: After you've created a customized user ID, you can copy its permissions and assign them to a new user ID. This is useful when you need to create multiple user IDs with the same permissions.

To copy permissions to a new user

- 1 Complete steps 1 and 2 above (To add a new user ID).
- 2 From the **User List**, select the user ID that has the permissions you want to reuse (for example, jsmith)
- 3 Click **Copy Permissions**.

The copied permissions appear, selected, in the **Roles** menu as the item **Like jsmith**.

To remove a user ID

- 1 Select it from the **User List**.
- 2 Click **Delete User**,
- 3 Click **OK**.

NOTE: You cannot delete the default user ID and any currently active user IDs.

To manage user signon-related actions

The last two entries in the **Actions** group manage the signon-related actions for the userid. They do not manage user permissions.

Actions > Unlock User

This option allows the console operator to end the userid's signon lockout period immediately.

Actions > Reset Passwd

This option allows the console operator to force the specified userid to change their password the next time they sign on.

Manage Password Security

The **Manage Password Security Parameters** dialog box contains several checkboxes and a few editboxes you can use to define the security rules for passwords and related security items.

Enforce Minimum Length

Checking this checkbox enables this rule. The associated edit box allows you specify the minimum length. This rule only applies when a console user sets or changes their password. It does not apply to preexisting passwords.

Require Numeric Characters, Require Alphabetic Characters, Require UC and LC Characters, and Require Special Characters

These checkboxes enable the corresponding rules for legal passwords. These rules only apply when a console user sets or changes his password. They do not apply to preexisting passwords.

If you want to force existing passwords to follow the above mentioned rules, use the **Manage Userid Permissions** dialog described in [“Manage User Permissions” on page 67](#) to force the corresponding users to reset their passwords.

The remaining entries in this dialog define the password validation rules.

Change Password on First Use

This checkbox forces a new userid to change their password the first time they sign on to the service by disabling all actions except changing their password. This makes the initial password a one-time password. After changing the password, the user must disconnect and connect again and signon using the new password.

Maximum Signon Retries

This entry is designed to prevent hackers from trying to guess passwords by constantly retrying. After the specified number of retries, it locks out the userid for the period of time specified by the value in the **Userid Lockout Timeout** (units are in minutes).

Set Default Password


When you create a new userid or reset the password of a userid, it assigns the default password to the user. Use **Set Default Password** to change this default password to prevent it from becoming common knowledge. If you do not set the default password, it defaults to an empty password.

On new systems, the default userid is dbridge. Use the default password in conjunction with the **Change Password on First Use** checkbox to help protect the default password. After a new user goes through the initial login process and changes their password, change the default password so it is different the next time you use it.

Change the Console Password

Use this procedure to change the password for your user ID. The Client Console requests this user ID and password after it connects to a service. The service manages user IDs and passwords in the service configuration file. By editing the service configuration file, the administrator can assign different privilege levels to user IDs. For details, see Appendix E in the *Databridge Client Administrator's Guide*.

To change the password

- 1 In the **Explorer** pane, right-click the service  and click **Change Password**.
- 2 Type your current password in the **Old Password** box and the new password in the remaining two boxes.
- 3 Click **OK**.

The password is encoded and saved in the service configuration file (`dbcontrol.cfg`).

7 Managing Operations

This section covers ways you can monitor and maintain your data sources after you clone the data source.

Stop a Process

From the Client Console, you can manually stop (and restart) a process that's currently running. You can also configure the data source to stop a process at specific times or when certain events occur.

To manually stop a single process command

- 1 In the **Explorer** pane, under **Runs**, right-click the process you want to stop and click **Stop**.
- 2 In the **Stop Process** dialog box, click one of the following, and then click **OK**:
 - ♦ To stop the client at the next quiet point, click **Normal**.
 - ♦ To stop the client after processing a particular audit file, click **AFN** and enter an audit file number.
 - ♦ To stop the client at a particular time, click **Time** and specify a date and time.
- 3 To resume processing, right-click the data source and click **Process** or let the Client start at the next scheduled run.

To stop a process automatically

Use this procedure to automatically stop client processing by using **Stop Conditions** parameters.

- 1 In the **Explorer** pane, right-click the data source and click **Client Configuration**.
- 2 Click the triangle next to **Processing** to expand the options and then click **Stop Conditions**.

For a description of these options, click the **Help** button  in the lower-left corner of the **Client Configuration** dialog box.

Schedule Updates

Use this procedure to configure the Client to check for updates at scheduled times or at specific intervals. Scheduling is a function of the Client Manager service. The service starts DBClient to run a `process` command at the specified times.

To schedule updates

- 1 In the **Explorer** pane, right-click the data source and click **Client Configuration** to open the Client Configuration dialog box.
- 2 On the left, click the triangle next to **Processing** to expand the options, then click **Scheduling**.

For a description of these options, click the **Help** button  in the lower-left corner of the **Client Configuration** dialog box.

- 3 Do one of the following:
 - ◆ To specify regular intervals, select **Fixed delay (in seconds)** and type a number.
 - ◆ To specify times of day, select **Daily** and then click a **<choose time>** button. In the analog clock that appears, drag the hour and minute hands to the time you want. Click PM in the center of the clock to toggle to AM. Click the green check mark to close the dialog box. Specify as many times as you want.
- 4 From the **Automatic Run Options**, select any additional options you want and then click **OK**.
Your selections are saved to the binary configuration file (`dbcontrol.cfg`) for the Client Manager service.

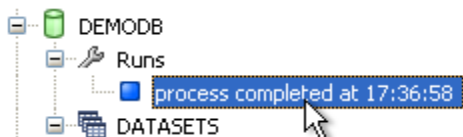
NOTE: If an unrecoverable error occurs, the data source in question and any scheduled updates for that data source are temporarily suspended. This saves significant mainframe resources that would otherwise be used in repeated failed attempts at running DBClient.

View Audit Information

You can get details about recently-processed audit files and other information related to a client run. The **Runs** list shows only the last ten runs for the data source. For information about older runs, refer to the log file (in the logs directory of the data source).

To view audit file information

In the **Explorer** pane, under **Runs**, right-click a completed process command and click **Client Status**.



The **Properties** pane opens. Audit file information is listed in the **Ending State Info** category.

Get Statistical Information

You can get performance and other statistical information about the server and the client any time during a `process` or `clone` command. After the command is terminated, you can view whatever statistics the Client Console collected. For example, you can get information about the server's processor, throughput times, and how many records it sent. For audit information, see [“View Audit Information” on page 72](#).

To get statistical information

- 1 In the **Explorer** pane, click the triangle next to the data source to expand it, then click the triangle next to **Runs** to expand it.
- 2 Right-click a process and click one of the following options:

NOTE: Some options are only available when the run is currently processing.

This option	Does this
Show DMSII Counters Chart	Shows counters based on incremental statistics that the client accumulates while processing audit files. These are shown after the current audit file is processed.
Show Process Lag Chart	The elapsed time between when an update is applied in the DMSII database and when the update appears in the relational database. The plotted value is the difference between the current time and the timestamp of the audit.
Show Task Time Chart	A pie graph that represents the time increments for each task the client performed when processing the current audit file.
Query Client Status (status)	Queries the current status of the Client and then displays status information, such as the current stateinfo during audit file processing, the trace status, and the program status (for example, waiting for TCP for 100 ms). The results appear in the Console pane.
Query Performance Statistics (pstats)	Queries the client for performance statistics on the current run by issuing a <code>pstats</code> command. The statistics that have accrued to that point are displayed in the form of a report in the Console pane.
Query Server Statistics (sstats)	Displays Databridge Server statistics in the Console pane at the next quiet point. The information includes I/O time, processor time, records sent, and priority.
Show Performance Statistics	Displays statistics in the Performance Statistics pane instead of the Console pane.
Export	Saves performance statistics to a comma separated value (<code>.csv</code>) file.

Set Up Service Log Files

Use this procedure to specify when a new log file will be created for the service. You can also manually start a new log file at any time.

The service log includes all service-related activity for the system and is located in the logs subdirectory of the working directory.

To set up log files

- 1 In the **Explorer** pane, right-click the data source and click **Client Configuration**.
- 2 In the **Client Configuration** dialog box, in the left pane, expand **Logging/Tracing**.
- 3 Below **Logging/Tracing**, click **Service Log** and select the settings you want. If you have multiple systems with services, specify a prefix to identify system-specific log files that would otherwise have the same name.
- 4 Click **OK**.

To manually start a new log file

- ♦ In the **Explorer** pane, right-click the service and click **Switch Service Log**.

Override the Databridge Engine COMMIT Frequency


From the Client Console, you can override the COMMIT frequency for the duration of the Client connection or only for the active `process` command.

The **Task Time** pane provides COMMIT statistics that can help you determine how much to adjust these parameters to improve performance. For more information, see [“Get Statistical Information” on page 72](#).

To change the Commit frequency

Use this procedure to temporarily change the Databridge Engine Commit frequency for the Databridge Engine or Databridge Enterprise Server as the default Client behavior. The change remains in effect only for the duration of the Client connection.

- 1 In the **Explorer** pane, right-click the data source and click **Client Configuration**. The Client Configuration dialog box opens.
- 2 On the left side of the Client Configuration dialog box, click the triangle next to **Processing** in the **Explorer** pane and then click **DBEngine and DBEnterprise Parameters**.
- 3 Use the **Commit parameters** to specify the type of Commit frequency you want the client to change.

For a description of these options, click the **Help** button  in the lower-left corner of the **Client Configuration** dialog box.

- 4 When you're done, click **OK**.

Your settings are saved to the client configuration file.

To override the Commit frequency for an active process command

Use this procedure to change the Databridge Engine Commit frequency for an active process command. Your settings apply to the active DBClient run in question and do *not* affect the default Commit frequency setting specified in the client configuration file.

- 1 In the **Explorer** pane, select a data source.
- 2 From the **Runs** menu, select **Client Options**.
- 3 In the **Client Options** dialog box, specify a value for one or more parameters.

Resynchronize the Service

If the Client Console displays outdated information, it has probably fallen out of sync with the Client Manager service. If this occurs after you have changed the filter in the configurator, back out to refresh the tree view. If simple actions like this fail, use the following procedure.

To reconnect the service

- 1 In the **Explorer** pane, right-click the service and click **Disconnect**.
- 2 Right-click the service again and click **Connect**.
- 3 Type the password (if any) and click **OK**.

The Client Console will reload information from the service.

Default Trace and Log Options Dialog Box

To open the **Trace and Log Options** dialog box, right-click on the data source and then click **Advanced**. Use the options in this dialog box to enable tracing.

Trace files are primarily used by Databridge Technical Support for diagnostic purposes. Trace files are not enabled by default because they use a significant amount of hard disk space.

We recommend that you use the default trace options when sending a trace to Micro Focus Support, unless you are specifically asked to use different options. You can also specify default trace options for a `process` or `clone` command by using **Data Source > Advanced > Process (with options)** (just check the **Enable default tracing** option in the dialog box).

For more information about trace files, see "Trace and Log Files" in the *Databridge Client Administrator's Guide*.

Global Selection Options

Select all	Click to select all of the trace options
Select default	Click to select the default trace options
Select none	Click to deselect all of the trace options
Current trace	Click to retain the currently set trace options (default)

Trace Options

Log output	Writes log messages (in addition to trace information) to the trace.log file. (Decimal value: 1 Hexadecimal value: 0x1.)
Server messages	Enables tracing of Databridge Server communications messages for key actions associated with Databridge on the host, including RPCs such as DB_SELECT and DB_READ and their responses. (Decimal value: 4. Hexadecimal value: 0x4.)
Database API calls	Enables the logging of calls from the Databridge Client to the ODBC or OCI APIs. This is also referred to as relational database API tracing. (Decimal value: 16. Hexadecimal value: 0x10)
TCP/IP data	Enables a protocol trace of the information exchange between Databridge Server on the host or Enterprise Server and the Databridge Client. The blocks of data are traced as they are read and written to the TCP interface. The messages are listed in DEBUG format, which is an offset followed by 16 bytes in hexadecimal, followed by the same 16 bytes interpreted as EBCDIC text. The non-printable EBCDIC characters are displayed as periods (.). (Decimal value: 64. Hexadecimal value: 0x40.)
Debug	Enables debugging code that may be added to engineering releases of the Databridge Client. (Decimal value: 256. Hexadecimal value: 0x100.)

Remote console RPCs	<p>Enables remote console message tracing when running DBClient or DBCIntCfgServer in conjunction with the service.</p> <p>(Decimal value: 1024. Hexadecimal value: 0x400.)</p>
Callback exit	<p>Enables a trace that helps determine if delays occurred while SQL statements were being executed or while the Databridge Client was waiting for data from the Databridge Server. (Decimal value: 4096. Hexadecimal value: 0x1000.)</p>
Verbose trace	<p>Enables the tracing of certain entries that are normally not traced as the information they provide is often of rather dubious value.</p> <p>(Decimal Value: 65,536. Hexadecimal value: 0x10000)</p>
DMS buffers	<p>Enables the tracing of the allocation and freeing of DMSII buffers. Use this trace when trying to debug a DMSII buffer leak.</p> <p>(Decimal Value: 262,144. Hexadecimal value: 0x40000)</p>
Buffer sizes	<p>Enables a trace that Shows the size calculations for the SQL and host variable buffers. This trace is only useful when you have problem with buffer or host variable overruns.</p> <p>(Decimal Value: 1,048,576. Hexadecimal value: 0x100000)</p>
SQL commands	<p>Enables a trace of SQL commands.</p> <p>(Decimal value: 2. Hexadecimal value: 0x2.)</p>
Control table load	<p>This is load tracing. Load tracing logs information on the Databridge Client control tables as they are loaded from the relational database.</p> <p>(Decimal value: 8. Hexadecimal value: 0x8.)</p>
Bulk loader data	<p>This is a trace of the records that are written to temporary data files (or UNIX pipes) and used by the bulk loader utility during the data extraction phase of cloning.</p> <p>(Decimal value: 32. Hexadecimal value: 0x20.)</p>
Remote console data	<p>This is a protocol trace of the information exchange between DBClient or DBCIntCfgServer and the service. The output looks like a Databridge Server protocol trace, except for the fact that all the data is ASCII.</p> <p>(Decimal value: 128. Hexadecimal value: 0x80.)</p>
List config file	<p>This option displays the configuration file parameters as they are processed.</p> <p>(Decimal value: 512. Hexadecimal value: 0x200.)</p>
User script SQL	<p>Temporarily enables SQL tracing when running user scripts during <code>define</code> and <code>redefine</code> commands.</p> <p>(Decimal value: 2048. Hexadecimal value: 0x800.)</p>
DOC Records	<p>Enables a trace of DOC records, which must be enabled. DOC records help get a better insight of what is going on in the audit files. This option is meaningless when Server Message tracing is enabled, because that trace includes DOC records.</p> <p>(Decimal Value: 8192. Hexadecimal value: 0x2000)</p>

Thread Trace	Enables a trace that shows when thread are started and stopped, this trace is seldom useful, unless you are dealing with hung threads. (Decimal Value: 131,072. Hexadecimal value: 0x20000)
Row Count	Enables a tracing of the row counts after SQL statement are executed. (Decimal Value: 524,288. Hexadecimal value: 0x80000)
Trace mask	This entry is automatically filled in as you select trace options. It represents the hex value of the trace mask that is added to the command line using the /t option using a prefix of 0x.
Log options	
Verbose	Logs some additional information. Most of the verbose output that you had in older versions is now available in the trace options.

Related Topics

- ♦ [“Set Up Service Log Files” on page 73](#)

8

Backing Up the Configuration

This section provides instructions for backing up your configuration settings.

For information about backing up the client control tables using the **Unload Data Source** command, see “[Disabling, Unloading, or Removing a Data Source](#)” on page 65.

Archive User Scripts

You can save your customizations to the relational database table layout as user scripts. If the system fails, the `define` command automatically uses these user scripts to rebuild the client control tables.

If you modify your relational database table layout in the Client Configurator, it's a good idea to save your changes as user scripts for backup purposes. If a hard disk or system failure occurs, you can use these user scripts to recover your customized table layout.

NOTE: You don't need to archive user scripts from previous versions of the Databridge Client. The Windows installer provides you with an option of copying the configuration files and user scripts from the previous version's working directory to the new working directory.

To set the user script directory

- 1 Create a directory for your user scripts in the location you want.
- 2 In the **Explorer** pane, right-click the data source and click **Client Configuration**.
- 3 On the left side of the **Client Configuration** dialog box, click the triangle next to **Customizing** to expand the options and then click **User Scripts**.
- 4 For **User script directory**, accept the default path or type a new path. The default is the scripts subdirectory in the working directory of the associated data source.
- 5 (Optional) For **User scripts archive directory**, specify the location where the user scripts archive directory is created. This directory uses the data source name followed by the date (and time, if multiple archives are created on the same day).
- 6 Click **OK**.

To create user scripts from the Client Console

- ♦ With the data source selected in the **Explorer** pane, click **Data Source > Advanced > Create User Scripts**.

After the Client copies the existing user scripts to the user scripts archive directory, it creates a new set of user script files in the user scripts directory.

User script archive directories aren't overwritten or deleted when new ones are created. Delete any that are no longer useful to you.

Export the Client Configuration to a File

Use this procedure to create a readable copy of the Client configuration file. The Client configuration file contains the current configuration settings for a data source, which are set from the Client Console and the Client Configurator. (The Client Console also includes settings to configure the service. These settings are saved to the service configuration file. For more information about this file, see "Client Configuration Files" in Appendix D of the *Databridge Client Administrator's Guide*.)

On rare occasions, you'll need to change a parameter that can't be set in either the Client Console or the Client Configurator. In this case, you'll need to export the configuration file, edit it, and then import the file.

To export the client configuration file

- 1 In the **Explorer** pane, right-click the data source and choose **Client Configuration**.
- 2 On the left side of the Client Configuration dialog box, click **Customizing**, and select **Encode passwords in configuration INI file**.
- 3 In the **Explorer** pane, right-click the data source and click **Advanced > Export Client Configuration**.

NOTE: Passwords in the Client configuration file are automatically encoded when you use the `export` command to export the file.

A new text data source configuration file (`dbridge.ini`) is created from the binary configuration file (`dbridge.cfg`).

To import the file, see "[Import a Client Configuration File](#)" on page 80,

Import a Client Configuration File

Use this procedure to create a binary configuration file from a text client configuration file.

- ♦ Open a command session and run the following command:

```
dbutility [options] import [filename]
```

where `[filename]` is an optional parameter to specify a filename other than the default (`dbridge.ini`). When no option or filename is specified, the `import` command processes the text file `dbridge.ini` in the config directory and creates an equivalent binary configuration file, `dbridge.cfg`, in the same directory. If the file `dbridge.ini` does not exist in this location, the `import` command creates a binary configuration file with the default values. If the text file contains errors, the Client returns an error to help you identify the problem and no binary file is created.

Option	Description
<code>-f filename</code>	Use this option to specify a filename or path other than the default. If this option is omitted, the Client tries to read the file <code>dbridge.cfg</code> in the config directory of the data source's working directory. To indicate a different location, type a backslash (Windows) or forward slash (UNIX) followed by the full path, including filename. For example, <code>/home/user/xyz/foo/myconfig.cfg</code>

Option	Description
-u	<p>This option is required to allow the existing configuration file to be overwritten with a new file with the same name. Otherwise, the Client will try to read (import) the configuration from a file named <code>dbridge.ini</code>.</p> <p>For example, the following command:</p> <pre>dbutility -u -f dbridge.cfg import</pre> <p>imports (reads) a file named <code>dbridge.cfg</code> and creates the binary configuration file <code>dbridge.cfg</code> regardless of whether the imported file is a text or binary file.</p>

Display the Client Control Tables

The client control tables do not contain replicated DMSII data. The control tables hold the layout information of the DMSII database (from the DMSII DESCRIPTION file) and the layout of the corresponding relational database tables. The DATASETS table contains state information about the replication process. To store replicated data, the relational database uses data tables, which are created using information from the control tables. Each relational database has one set of client control tables that includes the following tables: DATASOURCES, DATASETS, DATATABLES, DMS_ITEMS, and DATAITEMS. (The exception is a test environment where a second set of client control tables are created for testing purposes. We don't recommend this.)

The Databridge Client uses certain columns of the control tables to determine how DMSII database objects are represented in the relational database layout. While Databridge makes many of these decisions, some properties can be modified by using the Client Configurator or user scripts. For example, you can rename columns, combine like items, and flatten OCCURS, to name just a few.

You can verify changes you made to a single data set by viewing only the control table entries for that data set. This output shows changes made in the **Client Configuration** dialog box or the **DMSII** and **Relational Properties** panes of the Client Configurator.

To display the client control tables

- ◆ From **Explorer** pane, select a data set or a data source, and then do one of the following:
 - ◆ To display control tables on-screen in a Console pane, click **Data Source > Advanced > Display Control Tables**.
 - ◆ To write the control tables to the current client log file, click **Data Source > Advanced > Log Control Tables**.

For space considerations, column names are abbreviated to their "display" name. For a key to these abbreviations, see ["Key to Control Table Columns" on page 81](#). For a full description about the columns, see Chapter 6 of the *Databridge Client Administrator's Guide*.

Key to Control Table Columns

Use the following tables to interpret abbreviations used in client control tables displayed from the Client Console. For more information, see Chapter 6 in the *Databridge Client Administrator's Guide*.

DATASETS

ds/type	dataset_name/rectype	ST	subtype	AFN	audit_filenum
set	set_name	ICNT	item_count	ABSN	audit_block
A	active	OP	ds_options	SEG	audit_seg
ST#	strnum	MISC	misc_flags	INX	audit_inx
FMT	client_fmtlevel	EXTC	external_columns	Time Stamp	audit_ts
RSZ	recsz_bytes	VDS	virtual_ds_num	M	ds_mode
P#	parent_strnum	RDS/type	real_ds_num real_ds_rectype	SB	status_bits
#C	num_children	chg	changes	MAXRECS	max_records
B#	base_strnum	BMASK	ds_user_bmask	PRI	extract_priority

DATATABLES

ds/type	dataset_name/rectype	T#	table_number	isf	index_suffix
table_name	table_name	A	active	OP	dt_options
index	index_name	P	prim_table	chg	changes
occ	occurs_level	suf	create_suffix	BMASK	dt_user_bmask

DMS_ITEMS

item name	dms_item_name	S	dms_scale	Ink	dms_link_ds_num
A	active	OFF	dms_offset	ST	dms_subtype
K	item_key	LEN	dms_length	CAT	dms_concat_num
#	dms_item_number	s	dms_signed	OPTIONS	di_options
P#	dms_parent_item	#O	dms_num_occurs	OPT2	di_options2
T	dms_item_type	#D	dms_num_dims	chg	changes
DL	dms_decl_length	dep	dms_depends_num	BMASK	di_user_bmask
				BM2	di_user_bmask2

DATAITEMS

#	item_number	V	virtual_key	chg	changes
		K			
item_name	item_name	STY	dms_subtype	BMASK	da_user_bmask
A	active	OLV	occurs_level		
K	item_key	OPTIONS	da_options		

Abbreviations for columns grouped under DMS

DMS I#	dms_item_number	DMS SC	dms_scale	DMS OCC	dms_num_occurs
DMS P#	dms_parent_item	DMS OFF	dms_offset	DMS dep	dms_depends_num
DMS TYP	dms_item_type	DMS LEN	dms_length		
DMS DL	dms_decl_length	DMS s	dms_signed		

Abbreviations for columns grouped under SQL

SQL TY	sql_type	SQL LEN	sql_length	SQL SC	sql_scale
---------------	----------	----------------	------------	---------------	-----------

