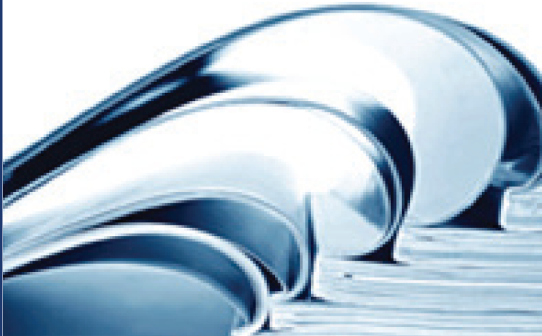
DataFlux® Data Management Server Administrator's Guide



sas®

DATAFLUX®
data management

This page is intentionally blank

DataFlux® Data Management Server

Administrator's Guide

Version 2.4

December 18, 2012

This page is intentionally blank

Contact DataFlux

DataFlux Corporate Headquarters
Toll Free: (877) 846-3589
Tel: (919) 447-3000
Fax: (919) 447-3100
940 NW Cary Parkway, Suite 201
Cary, NC 27513
USA

DataFlux West
Tel: (818) 906-7638
Fax: (818) 907-6012
15300 Ventura Boulevard, Suite 523
Sherman Oaks, CA 91403
USA

Technical Support

Phone: 919-677-8008
Email: techsupport@sas.com
Web: http://support.sas.com/techsup/contact/

Documentation Support

Email: yourturn@sas.com

Legal Notices

Table of Contents

Introduction

This section provides basic information about the DataFlux® Data Management Server (Data Management Server) product and documentation. Data Management Server supports all features available in the corresponding Data Management Studio release.

- Conventions Used in This Book
- DataFlux Reference Documentation

Accessibility

Data Management Server includes features that improve usability of the product for users with disabilities. These features are related to accessibility standards for electronic information technology that were adopted by the United States (U.S.) Government under Section 508 of the U.S. Rehabilitation Act of 1973, as amended.

If you have questions or concerns about the accessibility of DataFlux products, send an e-mail to techsupport@dataflux.com.

Conventions Used In This Document

This document uses several conventions for special terms and actions.

Typographical Conventions

The following typographical conventions are used in this document:

Typeface	Description
Bold	Signifies a button or action.
Italic	Identifies arguments or values that you supply, such as version numbers.
`Monospace`	Indicates filenames, directory paths, and examples of code.

Syntax Conventions

The following syntax conventions are used in this document:

Syntax	Description
#	The pound # sign at the beginning of example code indicates a comment that is not part of the code.
>	The greater than symbol is used to show a browse path. For example, **Start** > **Programs** > **DataFlux** > **License Manager *version***.

Path Conventions

Various products and operating systems may use different paths for default locations. This document uses the path for the 64-bit version of Microsoft Windows 7 in examples. The following examples display the differences in paths for three different operating systems:

Windows XP

> *drive*:\Program Files\DataFlux\DMServer\*(Undefined variable: MyVariables.instance-name)*

Windows 7

> 32-bit – *drive*:\Program Files (x86)\DataFlux\DMServer\*(Undefined variable: MyVariables.instance-name)*
>
> 64-bit – *drive*:\Program Files\DataFlux\DMServer\*(Undefined variable: MyVariables.instance-name)*

UNIX®

> $DATAFLUX_ROOT/dmserver

Reference Publications

This document may reference other DataFlux documentation, including:

DataFlux Migration Guide

DataFlux Authentication Server Administrator's Guide

DataFlux Authentication Server User's Guide

DataFlux Data Management Server User's Guide

DataFlux Data Management Studio Installation and Configuration Guide

DataFlux Data Management Studio User's Guide

Address Update Add-On to Data Management Studio Quick Start Guide

DataFlux Expression Language Reference Guide

DataFlux Federation Server Administrator's Guide

DataFlux Federation Server User's Guide

DataFlux Quality Knowledge Base Online Help

DataFlux Secure Administrator's Guide

DataFlux Web Studio Administrator's Guide

DataFlux Web Studio User's Guide

What's New in Data Management Server 2.4

The main enhancements for Data Management Server 2.4 include the following:

- Enhance the performance of Monitor jobs
- Enhance access control by IP address
- Browse available services and WSDLs
- Specify credentials when running jobs from a command line
- Persist and purge the histories of job run instances

Enhance the performance of Monitor jobs

Bulk loading is now used to enhance performance for Monitor jobs that include row-logging events. You can change the number of rows in each bulk load using the app.cfg configuration option MONITOR/BULK_ROW_SIZE. You can also specify that the Monitor store temporary jobs on disk using the app.cfg configuration option MONITOR/DUMP_JOB_DIR. To learn more, see Configure Bulk Loading in Monitor Jobs.

Enhance access control by IP address

Administrators can now allow or deny by IP address the capability of bypassing all security checks on the Data Management Server. See the new dmserver.cfg configuration option DMSERVER/IPACC/NOSECURITY.

Browse available services and WSDLs

You can now use a Web browser to display the services and WSDL files that are available on Data Management Servers. See Browse Available Services and WSDLs.

Specify credentials when running jobs from a command line

When you use a command line to run jobs, you can now specify a user name and password for authentication by an Authentication Server. See Run Jobs from a Command Line.

Persist and purge the histories of job run instances

You can now store the histories of job run instances so that they persist between restarts of the Data Management Server. See the new configuration option DMSERVER/JOBS_KEEP_HISTORY. Also, you can now specify when the server is to automatically remove the history items that are created when you run a job. See the new configuration option DMSERVER/JOBS_HISTORY_MAXAGE.

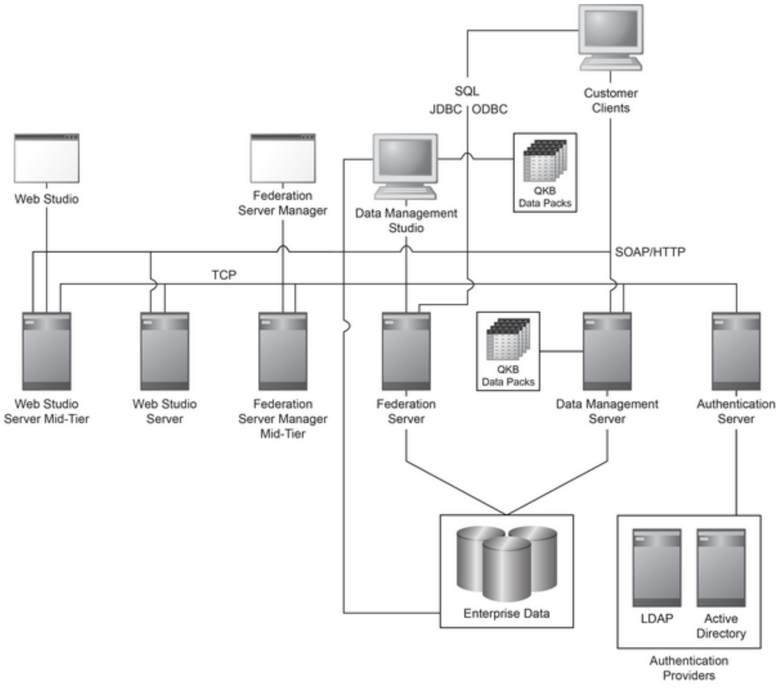
Introducing the Data Management Server

Overview

DataFlux® Data Management Server (Data Management Server) addresses the challenges of storing consistent, accurate, and reliable data across a network by integrating real-time data quality, data integration, and data governance routines throughout your IT environment. With Data Management Server, you can replicate your business rules for acceptable data across applications and systems, enabling you to build a single, unified view of the enterprise.

Working with DataFlux Data Management Studio, Data Management Server helps form the backbone of the DataFlux Data Management Platform. The server can implement rules created in Studio in both batch and real-time environments. Data Management Server enables pervasive data quality, data integration and master data management (MDM) throughout your organization.

The following figure illustrates the integration of Data Management Server in the DataFlux Data Management Platform:



Also included with Data Management Server is the ability to make Application Programming Interface (API) calls to the same core data quality engine. Discrete API calls are available through native programmatic interfaces for data parsing, standardization, match key generation, address verification, geocoding, and other processes. The Standard edition of Data Management Server requires a developer to code programmatic calls to the engine.

Data Management Server can be deployed on Microsoft Windows, UNIX, and Linux platforms with client/server communication using HTTP.

Configuration Options

Data Management Server reads configuration options from the **dmserver.cfg** configuration file. This file is created during installation and contains default values with the essential options to run Data Management Server. dmserver.cfg is located in the **etc** directory of the installation path, e.g.: *drive*:\Program Files\DataFlux\DMServer\*(Undefined variable: MyVariables.instance-name)*\etc\. . The file is in a key = value format and can be edited with any text editor.

For a complete list of configuration options, see the Configuration Options Reference for dmserver.cfg.

Editions of Data Management Server

Data Management Server is available in two editions—Standard and Enterprise. Data Management Server – Standard supports the ability to run batch Studio jobs in a client/server environment. The Standard edition allows any Studio client user to offload batch and profile jobs to a more scalable server environment. This capability frees up resources on client machines.

Data Management Server – Enterprise has increased functionality with the capability of calling business services designed in the Studio client environment. Additionally, Enterprise invokes real-time services.

About the Standard Edition

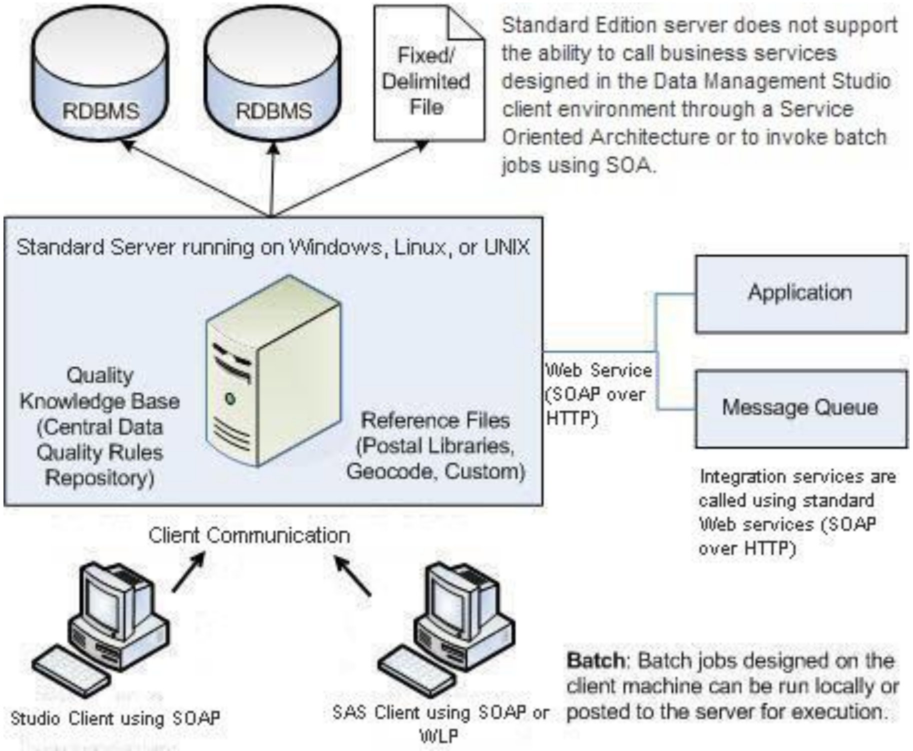
The Standard edition of Data Management Server supports native programmatic interfaces for C, C++, COM, Java, Perl, Microsoft .NET, and Web services. The server runs in its own process as a Microsoft Windows service or UNIX/Linux daemon. The Data Management Server installation includes both a client component and a server component, which communicate via Hypertext Transfer Protocol (HTTP) and SOAP over Transmission Control Protocol/Internet Protocol (TCP/IP).

Key Benefits

One key benefit of Data Management Server - Standard is that it supports the ability to run DataFlux Data Management Studio (Studio) batch jobs in a client/server environment by allowing users to offload Studio jobs onto a high-performance server.

Architecture

The following figure depicts integration architecture for Data Management Server – Standard.


RDBMS
RDBMS
Fixed/ Delimited File
Standard Edition server does not support the ability to call business services designed in the Data Management Studio client environment through a Service Oriented Architecture or to invoke batch jobs using SOA.
Standard Server running on Windows, Linux, or UNIX
Quality Knowledge Base (Central Data Quality Rules Repository)
Reference Files (Postal Libraries, Geocode, Custom)
Web Service (SOAP over HTTP)
Application
Message Queue
Integration services are called using standard Web services (SOAP over HTTP)
Client Communication
Batch: Batch jobs designed on the client machine can be run locally or posted to the server for execution.
Studio Client using SOAP
SAS Client using SOAP or WLP

About the Enterprise Edition

The Enterprise edition of Data Management Server offers an innovative approach to data quality that drastically reduces the time required to develop and deploy real-time data quality and data integration services. Through tight integration with the DataFlux Data Management Studio (Studio) design environment, Data Management Server – Enterprise operates as a data quality and data integration *hub*. Both batch and real-time services, which may include database access, data quality, data integration, data enrichment, and other integration processes, can be called through a service-oriented architecture (SOA). This eliminates the requirement to replicate data quality logic in native programming languages such as Java or C. Instead of writing and testing multiple lines of code, you can design the integration logic visually and then call from a single Web service interface.

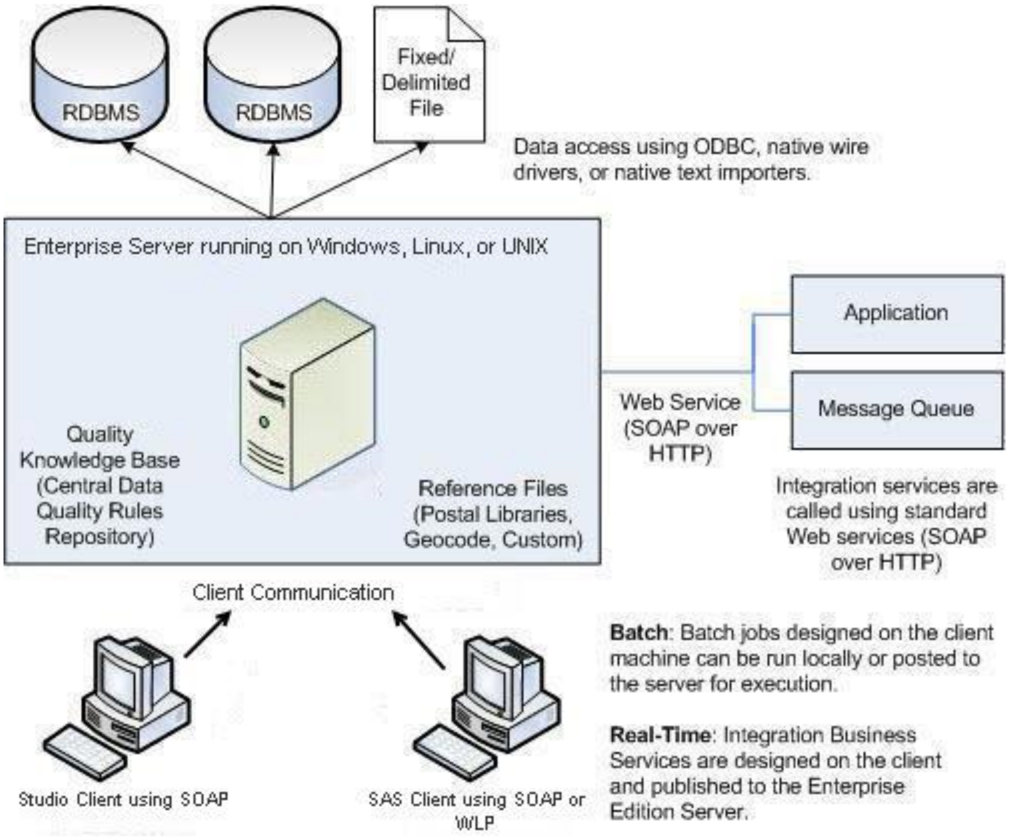
The Enterprise edition of Data Management Server supports real-time deployment using SOA, as well as the ability to run batch Studio jobs.

Key Benefits

One key benefit of Data Management Server- Enterprise, is that it supports the ability to run Studio jobs in a client/server mode by allowing users to offload Studio jobs onto a higher performance server. Another benefit is that it supports an SOA framework, enabling complete reuse of data quality and integration business logic.

Architecture

The following figure depicts integration architecture for the Enterprise edition of Data Management Server.


RDBMS
RDBMS
Fixed/ Delimited File
Data access using ODBC, native wire drivers, or native text importers.
Enterprise Server running on Windows, Linux, or UNIX
Quality Knowledge Base (Central Data Quality Rules Repository)
Reference Files (Postal Libraries, Geocode, Custom)
Web Service (SOAP over HTTP)
Application
Message Queue
Integration services are called using standard Web services (SOAP over HTTP)
Client Communication
Batch: Batch jobs designed on the client machine can be run locally or posted to the server for execution.
Real-Time: Integration Business Services are designed on the client and published to the Enterprise Edition Server.
Studio Client using SOAP
SAS Client using SOAP or WLP

Understanding the Processes of Data Management Server – Enterprise

The server is responsible not only for sending and receiving SOAP requests, but also for monitoring the progress of all registered data integration services. Once the server receives a request, the server sends the data to the invoked Web service. If the service has not been previously invoked, the server will load a new service process into memory and sends the data to it. If the service process invoked from the client application is busy, the server spawns a new service process and passes the data to the new service. Each service runs in its own process, which allows for robust error recovery, as well as the ability to spread the processing load across multiple CPUs. The server is always available and listening for additional client requests.

More specifically, the server handles the following processes:

- **Query server to return the names of available services**

  If the server receives a list services request, the server simply queries the services directory and returns the name of each found file.

- **Return input/output fields for a specified service**

  If the client queries the server for the input/output fields of a given service, the server sends the client the names and types of the existing input and output fields for that service.

- **Pass data and macros to a service, run the service, and receive output data and macros in return**

When the server receives a request to process data or macros from a client call, it identifies an idle service, sends the data to the idle service, and listens for additional client requests. If an idle service is not available, the server will load a new service into memory and pass the data or macros to the new service. The server monitors the service progress; as soon as the service returns output, the server sends the output back to the client application. If the service fails for any reason, the server will terminate the service process and return an error message to the calling application. After a service completes a request, both changed and unchanged data and macros will be reset to their default values.

Introducing the User Interface

The DataFlux® Data Management Server (Data Management Server) administration (user) interface is accessed from within Data Management Studio. To display the interface, open Data Management Studio and click the **Data Management Servers** riser bar. Studio responds by displaying a tree view of your Data Management Servers in the left-hand navigation pane. In the right-hand information pane, Studio displays a list of server names.

Using the Navigation Pane

The left-hand navigation pane provides a toolbar that contains the following icons:

Action Menu - Used to create, edit and delete a Data Management Server. Here you can change the server's credentials and unload idle processes (that are not real-time data services processes).

New - Used to register a new Data Management Server, so that you can connect to it.

Import- Allows you to import items from a repository.

Export - Allows you to export the selected object to a repository.

Edit - Allows you to export the selected object to a repository.

Expand - Allows you to expand all folders for the selected server.

In the tree view, you can expand a server to display information about the jobs and services that are available on that server. Right-click to connect.

Using the Information Pane

The right-hand information pane provides a toolbar that contains the following icons:

New - Allows you to register a new Data Management Server.

Edit - Allows you to import items from a repository.

Delete - Allows you to export the selected object to a repository.

Find - Allows you to edit the selected object. If this option is not available, you cannot edit the object.

Installing and Configuring the Data Management Server

Installation Notes

DataFlux Data Management Server is available through SAS delivery channels. See your SAS Software Order Email (SOE) for information about installing this product.

The default installation path under Windows is: SASHome\product-instance-name

The default installation path under UNIX is: SASHome/*product-instance-name*

In this document, the default installation path is indicate by the term *install-path*.

After you deploy the Data Management Server, refer to the other topics in this chapter to configure your server.

For information regarding the system requirements for the Data Management Server, refer to the SAS System Requirements page.

Install and Configure Additional Software

The data cleansing and data quality suite of applications encompassed by DataFlux Data Management Studio can be integrated into the service-oriented architecture of Data Management Server. This architecture can be customized to your own environment, using applications like dfIntelliServer, Quality Knowledge Bases (QKB), Accelerators, and DataPacks. For information on installing dfIntelliServer, QKB, Accelerators, refer to the relevant software installation documentation. For information on installing and configuring the Data Packs, including USPS, Canada Post, and Geocode, see the *DataFlux Data Management Studio Installation and Configuration Guide*.

Address Update Add-On

The DataFlux® Address Update add-on enables you to use the United States Postal Service (USPS) NCOALink® system to identify and update address information about customer records. For businesses and organizations with very large North America-based customer information databases, this feature is key for maintaining accurate and up-to-date address information for location-based marketing efforts, direct mail marketing, and similar activities.

Address update jobs can be imported from Data Management Studio to a Data Management Server, where the jobs are executed. One approach would be to run test jobs and small jobs on the Data Management Studio client workstation, and to upload larger jobs to the Data Management Server. Using this approach, both Data Management Studio and Data

Management Server must have all of the software, licenses, DSN connections, and other resources that are required to execute address update jobs and reports.

The following information outlines the necessary tasks associated with deployment of Address Update on the Data Management Server. For detailed information on installing and configuring Address Update see the *Address Update Add-On to Data Management Studio 2.2 Quick Start Guide* and *DataFlux Data Management Studio Online Help - Address Update Add-On, Using the Address Update Add-On with Data Management Server*.

Prerequisites

- Install and configure in Data Management Studio before performing the Data Management Server installation.
- Verify that you can run address update jobs and reports on the Data Management Studio client workstation. This is important because you will copy some resources from the Data Management Studio computer to the Data Management Server computer, and these resources should be validated before they are copied.
- Verify that users can connect to the Data Management Server that will be used to execute Address Update jobs.

Installation

- Download and install the CI 2011 Quality Knowledge Base on the server. Follow the QKB installation instructions that are supplied with the QKB software.
- Download and run the Address Update installer, *dmp22-server-addressupdate.win32.exe* on the Data Management Server, which updates the server to support the Address Update add-on.
- Install NCOALINK Data from the United States Post Office (USPS) on the server. Follow the instructions in the *Address Update Add-On to Data Management Studio Quick Start Guide* that is provided with the Address Update installer.
- Install USPS test data (CASS, DPV, and LACS) on the server. Follow the instructions in the *Address Update Add-On to Data Management Studio 2.2 Quick Start Guide*.

Configuration

Update the following in the **ncoa.cfg** file on the Data Management Server. **ncoa**.**cfg** is located in the **etc****macros** directory of your installation path.

- NCOA/DVDPATH – Where the USPS® NCOALink® data was installed.
- NCOA/QKBPATH – The location of the QKB. Minimum of CI 2011A is needed.
- NCOA/USPSPATH – Location of the USPS Address verification data.

The following are optional settings for the configuration of CASS/LACS and would also go in the **ncoa**.**cfg** file.

- NCOA/DFAV_CACHE_SIZE – Range: 0 through 100 and indicates how much data to cache. The higher the value the more data is cached, the faster the processing and the more memory used. The default is 0.

- NCOA/DFAV_PRELOAD – Set dfav/verify preload options. Provide the names of US states to preload, to speed up address verification for those states. Valid values:
  - "." - do not preload any states. This is the default.
  - "ALL" - preload all states.
  - "MIL" - preload military addresses only.
  - "CONUSA" - preload the 48 contiguous states.
  - "TX FL" - preload Texas and Florida

Add the following to the **app.cfg** file on the Data Management Server. You can copy these settings from the **app.cfg** in Data Management Studio as the information should be the same:

- NCOA/REPOSDSN - DSN connection for the address update repository.
- NCOA/REPOSPREFIX - Table prefix for the tables in this repository, if a prefix has been specified.
- NCOA/REPOSTYPE - Value that indicates the type of repository:

```
0 = NULL (the DataFlux Data Access Component (DAC) will
try to determine the repository type from the connect
string)
1 = ODBC DSN
2 = Custom DSN
```

After configuration, perform the following tasks to set up jobs on the Data Management Server. These items are explained in detail in *Data Management Studio Online Help, Using the Address Update Add-On with Data Management Server.*

- Create a separate Processing Acknowledgment Form (PAF) for the Data Management Server if Data Management Studio and Data Management Server are running on different operating systems. Reference topic: *Address Update Add-On, Administering the Address Update Add-On*, *Work with PAFs*.
- Enable jobs on Data Management Server to access an address update repository. Reference topic: *Enable Jobs on a Data Management Server to Access an Address Update Repository*.
- Configure a DSN on the Data Management Server identical to the DSN defined in the NCOA/REPOSDSN variable in Data Management Studio. Users will want to save credentials for this DSN. See Configure ODBC Connections for additional information.
- Import the Address Update Lookup jobs from Data Management Studio to the Batch Jobs folder on the Data Management Server. Then, you can execute the jobs on the server. Reference topic: *Deploy an Address Update Job to a Data Management Server*.

Quality Knowledge Base

If you are using the Quality Knowledge Base (QKB) on Data Management Server, the same version of QKB must be installed in Data Management Studio. Set the path to the QKB in the server's app.cfg file (in the etc directory.) Remove the comment character and replace the PATH value with a non-relative path.

```
# qkb/path = PATH
# Location of the active Quality Knowledge Base.
#
# example: qkb/path = C:\QKB
```

Configuring DataPacks

If DataPacks are installed on your Data Management Server, specify the paths in your app.cfg file.

CASS (US Data, USPS)

```
# verify/usps = PATH
# Location of US address verification data.
#
# example: verify/usps = C:\USPSData
```

Geocode

```
#
# verify/geo = PATH
# Location of Geocode/Phone data.
#
# example: verify/geo = C:\GeoPhoneData
```

SERP (Canadian Data)

```
# verify/canada = PATH
# Location of Canadian address verification data.
#
# example: verify/canada = C:\CanadaPostData
```

World

World Address Verification requires you to enter an unlock code in addition to the path. The unlock code is supplied with the DataPack.

```
# verifyworld/db = PATH
# Location of World address verification data.
#
# example: verifyworld/db = C:\Platon
#
# verifyworld/unlk = UNLOCK_CODE
# Unlock code provided by DataFlux for unlocking the World
address
 # verification functionality.
#
# example: verifyworld/unlk = ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Set Directory Permissions

The following tables outline the recommended permissions for users of Data Management Server.

Windows

Directories	Users	Default Permissions
install-path\DMServer	Administrator, Installer	Full Control
	Process user	Read and Execute, List Folder Contents
install-path\DMServer\var	Installer	Full Control
	Process user	Read, Write, List Folder Contents
	The user who backs up the Data Management Server, Backup Administrator	Read, List Folder Contents

UNIX

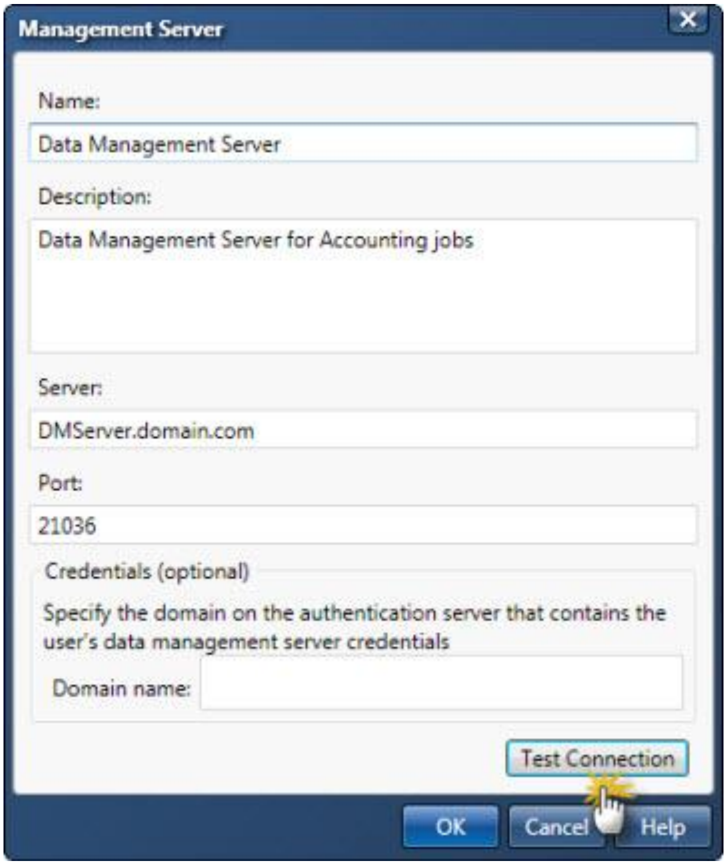
Directories	Users	Default Permissions
install-path/dmserver	Installer	Read, Write, Execute
	Process user	Read, Execute
install-path/dmserver/var	Installer	Read, Write, Execute
	Process user	Read, Write, Execute
	The user who backs up the Data Management Server; Backup Administrator	Read, Execute

Note: TMPDIR may have to be set in the event that the system's default temp directory (/TMP) runs out of space while running jobs or services. If this occurs, set the TMPDIR environment variable to read/write for the run-time user.

Register a Data Management Server

Follow these steps to register a Data Management Server in Data Management Studio:

1. Click to select the **Data Management Servers** riser bar to open the Data Management Servers interface.
2. Click **New Data Management Server** on the toolbar. The **Management Server** dialog opens.
3. In the **Management Server** dialog,
   - Specify a name for the server in the **Name** field.
   - Enter a description for the server.
   - Enter the server host name in the **Server** field.
   - Port **21036** is the default port number for Data Management Server. If using a different port, enter the number in the **Port** field.
   - Enter the domain name for the associated Authentication Server.



Management Server

Name: Data Management Server

Description: Data Management Server for Accounting jobs

Server: DMServer.domain.com

Port: 21036

Credentials (optional)

Specify the domain on the authentication server that contains the user's data management server credentials

Domain name:

4. Click **Test Connection** to verify that you can connect to the server. Click **OK** to close the Test dialog.
5. Click **OK** to close the Management Server dialog.

The new Data Management Server appears in the left navigation pane of Studio and is ready to be configured.

Edit a Server

To edit a server:

1. Click **Data Management Servers** in the navigation pane to display a list of servers in the information pane.
2. Select the name of the server you want to edit and click **Edit** on the toolbar.
3. In the **Edit Management Server** dialog, make the desired changes.
4. To test the connection to the server, click **Test Connection**. If you are prompted, log on to the server by entering your user ID and password, and then click **Log On**.
5. Click **OK**at the Test Connection dialog, and click **OK** to exit the Edit Management Sever dialog.

Delete a Server

To delete one or more servers:

1. Click **Data Management Servers**. In the information pane, click the name of one or more servers you want to delete.
2. Click **Delete** ✕.
3. Confirm that you want to delete the servers by clicking **Yes**.

   Note: Deleting a server instance in Studio does not remove the actual Data Management Server from the network.

Configure the Server to Execute Studio Jobs and Services

You create and test jobs and real-time services in Data Management Studio. You then upload those jobs and services to a Data Management Server.

To run a new job or service, the configuration of the Data Management Server needs to replicate the configuration of the Studio client. Certain nodes require specific option settings. Other nodes require additional configuration on the server, such as the creation of a repository.

Because jobs are created and configured in Data Management Studio, the documentation for how to configure both the client and server is provided in the *Data Management Studio User's Guide* and in the *Data Management Studio Installation and Configuration Guide*. The confiiguration information refers to the Studio client, but the configuration process also needs to be applied to the Data Management Server.

To run a particular job on the Data Management Server, you might need to:

- Configure a repository.

- Configure the Java plug-in.
- Set configuration options in the Data Management Server app.cfg file. A listing of app.cfg options is provided in the Data Management Studio User's Guide.
- Set configuration options in the Data Management Server configuration file dmsserver.cfg (see Configuration Options Reference for dmserver.cfg.)

In addition to transferring the Studio configuration to the Data Management Server, you also need to consider the application of access controls to specify the users who will be permitted to run a particular job or real-time service.

Administering the Data Management Server

- Start and Stop the Data Management Server
- Administer Log Files

Start and Stop the Data Management Server

- Windows
- UNIX/Linux

Windows

Start and stop the service using the MMC or the Control Panel > Administrative Tools.

1. Click **Start** > **Control Panel.**
2. Double-click **Administrative Tools** > **Computer Management**. .
3. Expand the **Services and Applications** folder.
4. Click **Services**.
5. Click **DataFlux Data Management Server**.
6. Click either **Stop the service** or **Restart the service**.

Note: You can also access the DataFlux Data Management Server service using Start > All Programs > DataFlux.

Modifying the Windows Service Log On

When Data Management Server is installed, the DataFlux Data Management Server service is started using the local system account. Because this account may have some restrictions (such as accessing network drives) it is suggested that you modify the service properties to have the service log on using a user account with the appropriate privileges, such as access to required network drives and files. For security reasons, you should assign administrative privileges only if necessary.

To modify the Data Management Server log on:

1. Select **Control Panel** > **Administrative Tools**.
2. Double-click **Services**, and select the **DataFlux Data Management Server** service.
3. Select the **Log On** tab, select **This account**, and enter **Account** and **Password** credentials for a user with administrative privileges.

UNIX/Linux

Start and stop the daemon using the dmsadmin application included in the installation. This application can be run using the command-line command: **./bin/dmsadmin *your_command*** from the installation root directory, where ***your_command*** should be one of the following:

Command	Description
start	Allows you to start the Data Management Server. For example: `./bin/dmsadmin start`
stop	Allows you to stop the Data Management Server. For example: **./bin/dmsadmin stop**
status	Allows you to check whether the Data Management Server is running.
help	Allows you to display help information.
version	Allows you to display the version information.

Administer Log Files

- Server Logs
- Data Service Logs
- Batch Job Logs

Server Logs

Every request to the Data Management Server is assigned a unique request identification that is logged into a server log file, **dmserver.log**. As the server is processing a request, every logged message corresponding to that request will start with the associated request identification (rid) with an exception for security messages that are logged differently.

An example of the default installation path to the server logs is:

- *install-path*\var\server_logs\log-subdirectory

With each new run instance, Data Management Server creates a log subdirectory in the server_logs directory. Each subdirectory is given a unique name, for example:

- Directory Name: **20110804-14.26-pid5072__034C24**
- Where **20110804** is the date that the run started, **14.26** is the time, **pid5072** is the OS process id, and __**034C24** is a unique log id.

You can change the behavior of log directories with the following configuration options in the server configuration file, dmserver.cfg:

Configuration Option	Usage
DMSERVER/WORK_ROOT_PATH = <path>	The path to the directory where the server creates its working files and subdirectories. To change the destination directory, enter a new path. The default installation path/directories are shown above.
DMSERVER/NO_WORK_SUBDIRS = No	Controls whether or not each server run creates a new log subdirectory. The default is **No** which specifies that all log and work files are created in subdirectories in the server_logs directory. To disable creation of the subdirectories change this value to Yes.

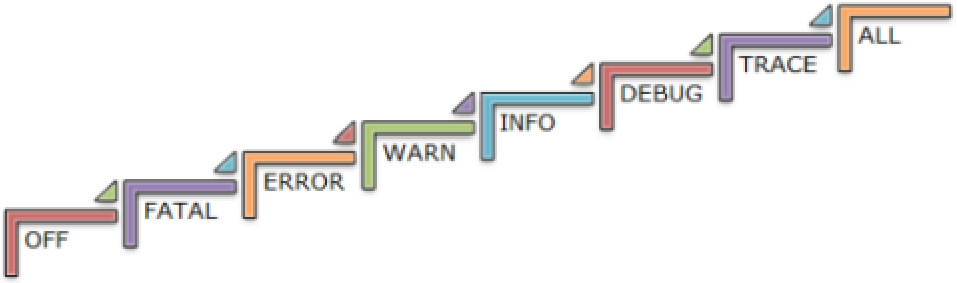
The level of server logging is controlled with the log configuration file **dmserver.log.xml**, located in *install-path*\etc.

Appenders and Loggers

As shown in dmserver.log.xml, the default log configuration consists of one appender and one logger. The appender specifies a log output destination. The loggers specify log event types and thresholds.

Loggers define the log events that are monitored. Loggers also define a threshold level for each monitored log event. The threshold levels determine the amount of information that is recorded in the log for each event.

Following is a list of threshold levels, ordered from the left as the least-information, to most-information at the right:



Log Events and Thresholds

The default log configuration captures most of the events that you will need to diagnose server problems. However, should there be a need to increase logging events and threshold levels, DataFlux Technical Support can assist with these configuration changes. Altering threshold levels above INFO when the server is operational in a production environment is discouraged since this may result in a reduction in server performance. All changes require a restart of the Data Management Server service.

Server Log Configuration (dmserver.cfg)

The path to dmserver.log.xml is set in dmserver.cfg. Following is the default value for dmserver.log:

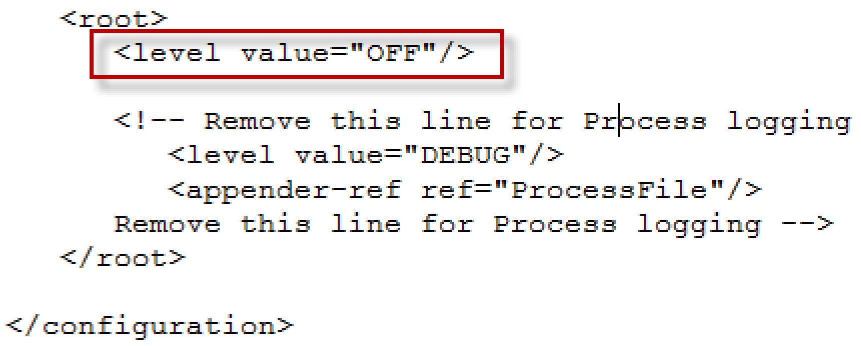
```
# base/logconfig_path = PATH
# Path to logging configuration file
#
# example: base/logconfig_path = C:\server\dmserver.log.xml
base/logconfig_path = install-path\etc\dmserver.log.xml
```

Data Service Logs

Data Services logging is controlled with **service.log.xml** located in *install-path*\etc. At the bottom of the configuration file is a block of text that controls *root logging*. The level value is set to OFF by default which means that logging is disabled to improve real time service performance. To enable services logging,

1. Open **service.log.xml** and locate the block of text that controls root logging.
2. Change the **OFF** value in <level value="OFF"/> to **DEBUG** or **TRACE**, depending on the level of information you want to gather.
3. Restart the Data Management Server.

Following is an extract of the default root logging parameters from service.log.xml:

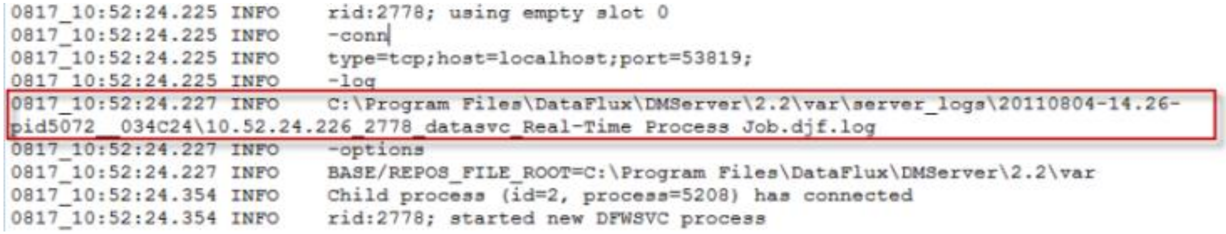


```
    <root>
      <level value="OFF"/>

      <!-- Remove this line for Process logging
         <level value="DEBUG"/>
         <appender-ref ref="ProcessFile"/>
      Remove this line for Process logging -->
    </root>

</configuration>
```

When data services logging is enabled, any request that starts a new **dfwsvc** run includes the name of a corresponding service log file that is shown in the dmserver.log. The server log also contains additional debug information associated with the service run. If a new service process is used, the server log includes the PID of the corresponding process.



```
0817_10:52:24.225 INFO    rid:2778; using empty slot 0
0817_10:52:24.225 INFO    -conn
0817_10:52:24.225 INFO    type=tcp;host=localhost;port=53819;
0817_10:52:24.225 INFO    -log
0817_10:52:24.227 INFO    C:\Program Files\DataFlux\DMServer\2.2\var\server_logs\20110804-14.26-
pid5072__034C24\10.52.24.226_2778_datasvc_Real-Time Process Job.djf.log
0817_10:52:24.227 INFO    -options
0817_10:52:24.227 INFO    BASE/REPOS_FILE_ROOT=C:\Program Files\DataFlux\DMServer\2.2\var
0817_10:52:24.354 INFO    Child process (id=2, process=5208) has connected
0817_10:52:24.354 INFO    rid:2778; started new DFWSVC process
```

The name of the file includes a time when the process started, request id (rid:000;) from the server log , service type, and service job name. Using the example:

- Server log entry: **10.52.24.226_2778_datasvc_Verify-Address-Job.ddf.log**
- Where **10.52.24.226** is the timestamp, **2778** is the request ID from the server log, **datasvc** is the service type, and the remaining information is the job log name **Real-Time Process Job.djf.log**.

If additional information is needed, contact DataFlux Technical Support for assistance to enable further logging.

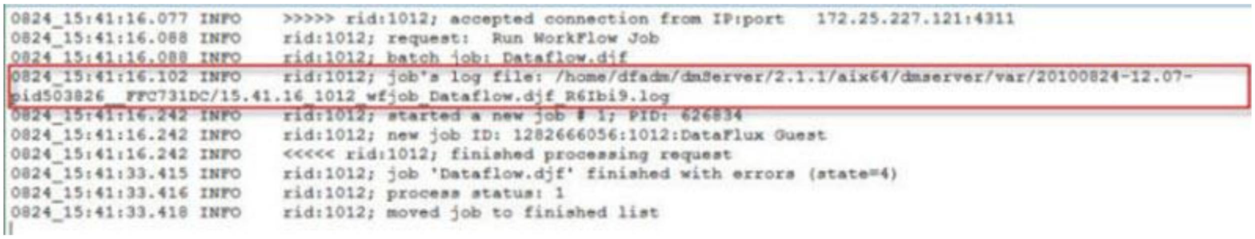
Services Configuration File (service.cfg)

This file is used to configure options specific to the execution of real-time services. It is also used to set the path to the service.log.xml file. The default is set at:

```
#
# base/logconfig_path = PATH
# Path to logging configuration file
#
# example: base/logconfig_path = C:\server\service_log.xml
base/logconfig_path = C:\Program
Files\DataFlux\DMServer\instance-name\etc\service.log.xml
```

Batch Job Logs

For every batch job run, Data Management Server tells the DFWFPROC process to create a log specifically for that job instance and logs the name and path to the log file in dmserver.log. As shown in the following example from release 2.1.1, the log file is **`15.41.16_1012_wfjob_Dataflow.djf_R6Ibi9.log`** with the path to that log file preceding the entry:



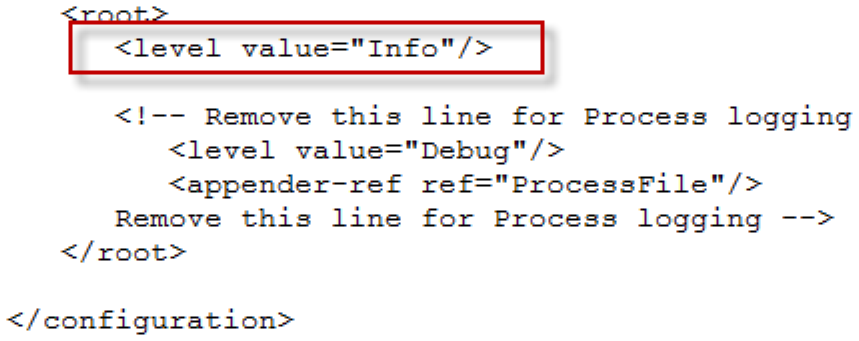
```
0824_15:41:16.077 INFO    >>>>> rid:1012; accepted connection from IP:port   172.25.227.121:4311
0824_15:41:16.088 INFO    rid:1012; request:  Run WorkFlow Job
0824_15:41:16.088 INFO    rid:1012; batch job: Dataflow.djf
0824_15:41:16.102 INFO    rid:1012; job's log file: /home/dfadm/dmServer/2.1.1/aix64/dmserver/var/20100824-12.07-
pid503826__FFC731DC/15.41.16_1012_wfjob_Dataflow.djf_R6Ibi9.log
0824_15:41:16.242 INFO    rid:1012; started a new job # 1; PID: 626834
0824_15:41:16.242 INFO    rid:1012; new job ID: 1282666056:1012:DataFlux Guest
0824_15:41:16.242 INFO    <<<<< rid:1012; finished processing request
0824_15:41:33.415 INFO    rid:1012; job 'Dataflow.djf' finished with errors (state=4)
0824_15:41:33.416 INFO    rid:1012; process status: 1
0824_15:41:33.418 INFO    rid:1012; moved job to finished list
```

Debug-Level Job Log

The root level value in batch job logging is set at Info by default. To troubleshoot an issue, you can change the logging level to debug by editing batch.log.xml located in install-path\etc.

1. Open **batch.log.xml** for editing and locate a block of text at the bottom of the log that controls *root logging*.
2. Change the "**Info**" value in <level value="Info"/> to "**Debug**" or "**Trace**" (Trace provides more details than debug) as shown in the example below.
3. Restart the Data Management Server service when you are finished.

Following is an extract of the default root logging parameters from batch.log.xml:



```
<root>
    <level value="Info"/>

    <!-- Remove this line for Process logging
       <level value="Debug"/>
       <appender-ref ref="ProcessFile"/>
    Remove this line for Process logging -->
</root>

</configuration>
```

If additional logging is needed, contact DataFlux Technical Support for assistance.

Batch Log Configuration File (batch.cfg)

Set the path to batch.log.xml file using the batch.cfg file. The default is set at:

```
 # base/logconfig_path = PATH
# Path to logging configuration file
#
# example: base/logconfig_path = C:\server\batch_log.xml
base/logconfig_path = install-path\etc\batch.log.xml
```

This file is also used to configure options specific to the execution of batch jobs.

Manage Security

Security Overview

The Data Management Server supports the following levels of security:

- **Unsecured** the default security mode after installation, grants access to all users to all of the Data Management Server's jobs, services, and data sources.
- **Secured by IP Address** grants access to server resources based on the IP addresses of the computers that connect to the server. This security level can be used in combination with the other levels of security.
- **Secured by Local Authorization** uses internal user and group definitions to authorize access to server resources, without using an Authentication Server.
- **Secured by Authentication Server** uses a central server to authenticate user credentials and provide group membership information.
- **Secured by SSL and AES** upgrades SOAP communication from HTTP to the Secure Sockets Layer (HTTPS). Encryption on disk and over the network is upgraded from the SASPROPRIETARY algorithm to the 256-bit AES algorithm. For further information, see Upgrade to SSL and AES

When security is not enabled, and if the Data Management Server requests authentication or group membership information from an Authentication Server, a SOAP fault message is returned stating that the request is disabled. Additionally, when security is disabled, the DataFlux Federation Server cannot be used and all data source names (DSNs) needed by jobs and services must be defined on the Data Management Server.

To control access by IP address, you set configuration options to specify the IP addresses that are allowed or denied the ability to submit SOAP commands or to post or delete objects on the server. To learn more, see Access Control by IP Address.

When your Data Management Server uses an Authentication Server, the servers manage security as follows:

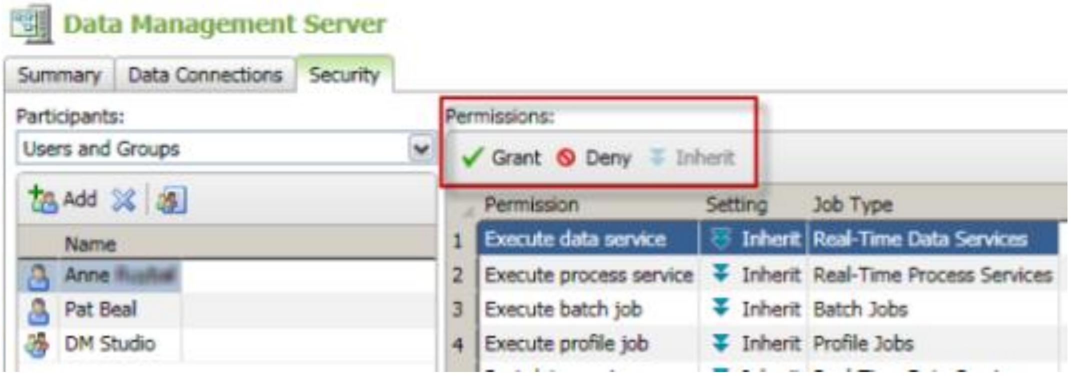
- About Authentication
- About Authorization

About Authentication

Authentication is the process of confirming the identity of users. When authentication is enabled on a Data Management Server, that server requires a connection to an Authentication Server.

When the Data Management Server receives a connection request, it passes credentials from the request to the Authentication Server.The Authentication Server sends the credentials to the authentication provider in the network domain that is specified in the credentials. If the user successfully authenticates, then the Authentication Server notifies the Data Management Server. The Data Management Server then establishes the user's command permissions by checking the **users** file.

If a user does not have specific command permissions already set, all of the commands permissions will default to *inherit*. This is the case when setting up new users. Unless specific permissions are granted, all of the permissions are set to *inherit* as shown in the following example. All new users' permissions are set at *Inherit* until they are updated.



Participants: Users and Groups

Name
Anne [redacted]
Pat Beal
DM Studio

	Permission	Setting	Job Type
1	Execute data service	Inherit	Real-Time Data Services
2	Execute process service	Inherit	Real-Time Process Services
3	Execute batch job	Inherit	Batch Jobs
4	Execute profile job	Inherit	Profile Jobs

Data Management Server Security: New User with Inherit Permissions

PUBLIC User

In Data Management Server, a PUBLIC user is a generic category that includes all users that do not exist in the Authentication Server. PUBLIC users do not have a unique Authentication Server ID. All such users are the same to the Data Management Server and fall into the PUBLIC users category. Data Management Server administrators can set command permissions for the PUBLIC group, just as they do for any Authentication Server admin-created group.

About Authorization

The following topic explains how Data Management Server performs authorization checks before granting access to a user or group.

Group Authorization Checks

Data Management Server checks group authorization in the following sequence:

1. The Data Management Server administrators group,
2. the DENY Group, if it is configured for use, and
3. the ALLOW Group, if it is configured for use.

Note: All groups must be created on the Authentication Server before they are configured on the Data Management Server.

Authorization starts when the Data Management Server checks if a user is a member of the Data Management Server administrators group. If a user is a Data Management Server administrator, no further authorization checks are performed and the user gets access to all Data Management Server commands and objects, regardless of specific permission and ACL settings.

Next, the Data Management Server checks if the deny group is configured. This group is set using the configuration option in **dmserver.cfg** located in [*drive*]:\Program Files\DataFlux\DMServer\*(Undefined variable: MyVariables.instance-name)*]\etc. If it is, and the user is a member of that group (directly or indirectly), no further authorization checks are performed and the user will be denied access to all Data Management Server commands and objects. Then, the Data Management Server checks if the allow group is configured. This can be set using the **DMSERVER/SECURE/GRP_ALLOW** configuration option in **dmserver.cfg**. If it is and a user is NOT a member of that group (directly or indirectly), no further authorization checks are performed and the user will be denied access to all Data Management Server commands and objects.

The allow and deny groups are optional and can be used as a convenient way to exclude sets of Authentication Server users from any access to Data Management Server without having to set specific permissions for them in specific Data Management Server installations.

After group memberships are checked, the Data Management Server looks at the following command permissions that are set for the user:

- If, for a given command, the user has *deny* set, the user is denied access. If an ACL exists, it will not be checked.
- If the user has *inherit* set, authorization checks proceed to group permissions . For more information, see Group Permissions.
- If the user has *allow* set, then the user gets access at this point if the request is not for a specific object. If there is a specific object involved in the request, the authorization checks then proceed to checking object's ACL. For more information, see ACL Authorization Checks.

Group Permissions

Group permissions are handled in accordance with the group's membership hierarchy. For example, a user can be a member of groups G1 and G2. Group G1 is a member of group G3. So, G1 and G2 are one step away from the user, and G3 is two steps away from the user. The authorization process looks at permissions on all group sets in an increasing order of steps from the user. If a command permission value can be determined from the groups that are one step from the user, Data Management Server will not look at permissions on the groups that are two steps from the user. When looking at a set of groups within the same distance from the user, if any group has *deny* permission for the command in question, the user is denied access. Otherwise, if any group has *allow* permission, then if there is an ACL to check, the authorization process moves to the ACL. Otherwise, the user gets access. If no group has specific permissions set, or the permission in question is set to *inherit*, authorization checks move to the set of groups one step further from the user.

If access rights cannot be determined after going through the regular groups (groups created in the Authentication Server) of which the user is a member, the next group whose permissions are checked is the USERS group. All users that the Authentication Server knows of and that are not public, belong to the USERS group. The Data Management Server admin can set command permissions and use it in ACLs just like any regular group. Access rights based on command permission for USERS group are calculated in the same way they are for other groups.

If access rights have not been determined, based on command permissions, the last step is for the Data Management Server to check whether permissions are set for the PUBLIC group. If the permission is *allow* and there is an ACL to check, the authorization check moves to the ACL. Otherwise, the user is granted access. If the permission is *deny*, *inherit*, or not set, the user is denied access.

If neither the user, any of groups of which the user is a member, the USERS group, or the PUBLIC group have permission set to allow access to a given command, the Data Management Server will deny access without checking the ACL. This means the Data Management Server requires a command permission that specifically allows access to a command, before Data Management Server will look at the ACL of an individual object, if one exists.

ACL Authorization Checks

Authorization checks of ACLs are similar to those performed for groups, except it is first checked whether the user is the owner of the object. If the object is owned by a group, the user must be a direct or indirect member of that group to be treated as object's owner. If the user is found to be the owner, no further authorization checks are done and the user is granted access to the object.

Next, ACEs are checked to see if they allow or deny permissions for the user. If nothing is found, ACEs are checked for groups of which the user is a member, taking into account the group's membership hierarchy as explained in the Group Permissions section.

If the ACL does not grant user access to the corresponding job or service, the user is denied access.

Enable an Authentication Server

Follow these steps to use an Authentication Server to authenticate users and centrally manage logins, users, groups, and shared logins.

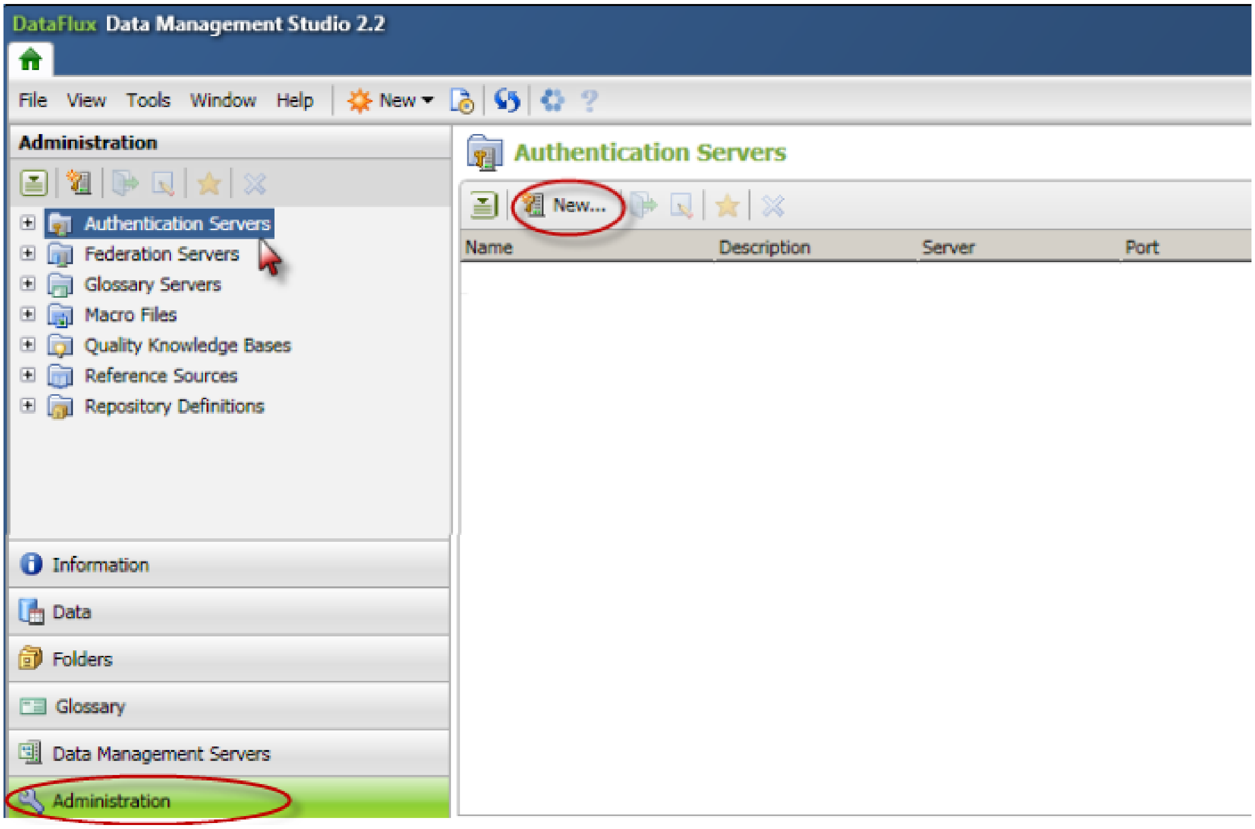
> **Note:** All communication between the Data Management Server and the Authentication Server is encrypted.

1. Register the Data Management Server in Data Management Studio. See Register a Data Management Server.
2. Install and configure an Authentication Server. See the *Authentication Server Administrator's Guide* for further information.
3. Create users, groups and logins on the Authentication Server.
4. Configure the Data Management Server to work with the Authentication Server.

Register an Authentication Server

The Authentication Server must be installed and visible on the network before it can be added to Data Management Studio. Refer to the *Authentication Server Administrator's Guide* for additional information. Once the Authentication Server is installed, create an instance in Data Management Studio as follows:

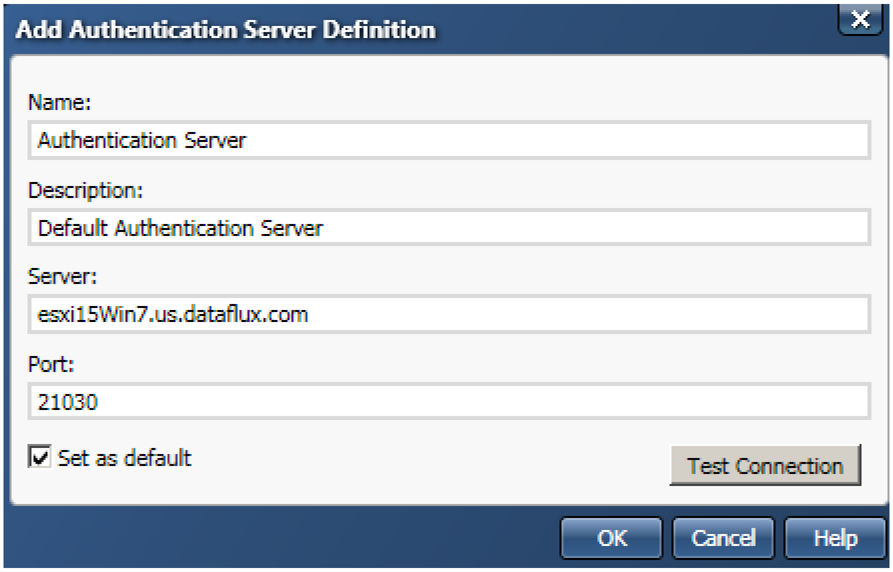
1. Open Data Management Studio and click the **Administration** riser bar.
2. Select Authentication Server and select **New...**.



3. At the *Add Authentication Server Definition* dialog, enter the server information and place a checkmark at **Set as default** if this will be the default Authentication Server.

   Note: Specifying a default Authentication Server controls the authentication method used for logins. If your environment is using one Authentication Server, it is designated as the default during installation.

4. Click **Test Connection** and click **OK** twice to close the dialog.



Name:
Authentication Server

Description:
Default Authentication Server

Server:
esxi15Win7.us.dataflux.com

Port:
21030

☑ Set as default

5. While logged into the Authentication Server, define the Data Management Server administrator group and add an administrator (user).
   - To add a new group, open the **Groups** riser, click **All Groups** and click **New Group**. Create the administrator group for the Data Management Server. e.g. DMServer_Admins.
   - To add a new user who will be the Data Management Server administrator, open the **Users** riser and click **New User.**
   - To add the Data Management Server administrator to the new group, open the **Groups** riser, click the new DM Server administrators group, e.g. *DMServer_Admins*, and click **Add Members.**

   ⚠ **Caution**: When creating the Data Management Server Administrators group that is subsequently set in DMSERVER/SECURE/GRP_ADMIN, keep in mind that all members of this group (whether direct members, or members of other groups that have been added to the admin group) will automatically have access to all commands and objects on the Data Management Server regardless of any other security-related settings.

Configure the Data Management Server to use the Authentication Server

After you add the Authentication Server and set up the administrators group and user objects, use the following procedures to configure the Data Management Server.

1. Edit **dmserver.cfg** and set the following configuration options:

   Note: The **dmserver.cfg** file is located in *install-path*/etc for UNIX systems.

   a. Enable security: **dmserver/secure = yes**

b. Configure the Data Management Server administrator group name in the following string: **dmserver/secure/grp_admin = (*name of admin group defined in Authentication Server*)** e.g. dmserver/secure/grp_admin=DMServer_Admins.

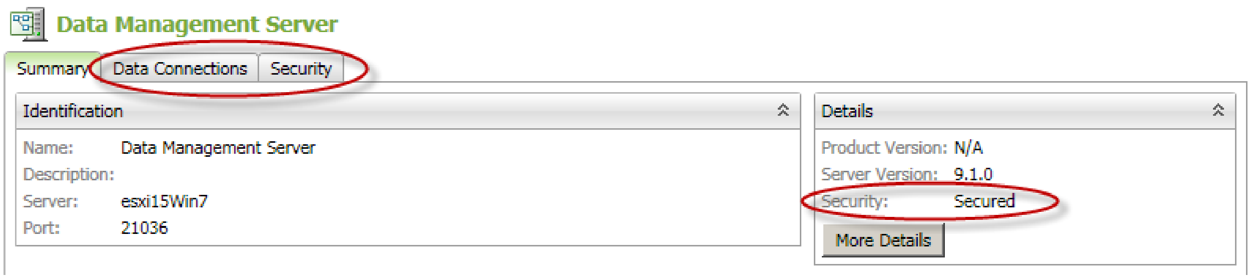
c. Point the Data Management Server to the Authentication Server using the fully qualified server name: **base/auth_server_loc = iom://*authserv.mycompany.com*:21030** (Port 21030 is the default).

Following is an example of the **dmserver.cfg** reflecting the changes outlined above. The edited configurations are presented in bold type:

```
#dmserver/secure=(yes|no)
# Enable the DIS security subsystem. If this is enabled clients will need
# be authenticated by the server before accessing any resources.
dmserver/secure = yes
dmserver/secure/grp_admin = DMServer_Admins

#
# base/auth_server_loc = URL
# Location and connection parameters for the Authentication Server.
#
# example: base/auth_server_loc = iom://authserv.mycompany.com:21030
base/auth_server_loc = iom://esxi15win7.us.dataflux.com:21030
```

2. Restart the Data Management Server by stopping and restarting the service.
3. Open Data Management Studio and log in if you are prompted.
4. Select the Data Management Server riser and open the Data Management Server that was just secured.
5. Log into the Data Management Server and notice, in the Details pane, the security status has changed to **Secured**. When the server is secured, the **Data Connections** and **Security** tabs are enabled.



Name: Data Management Server
Description:
Server: esxi15Win7
Port: 21036

Product Version: N/A
Server Version: 9.1.0
Security: Secured

Data Management Server Secured

Now that security is enabled, you can set up users and grant them permissions on the Data Management Server. All users and groups must be created on the Authentication Server before they can be accessed on the Data Management Server. See Manage Permissions for additional information.

Implement a Security Policy

As a resource on your network, Data Management Server usage can be defined based on your security model. The security model is based on usage policy, risk assessment, and response. Determining user and group security policies prior to implementation helps you minimize risk and expedite deployment.

> **Risk Assessment** — Security policies are inevitably a compromise between risk and necessary access. Users must access the application and data in order to perform necessary tasks, but there is associated risk when working with information, particularly confidential data. Consider the risks of compromised (unauthorized views or lost) data. The greater the business, legal, financial, or personal safety ramifications of compromised data, the greater the risk.
>
> **Usage Policy** — Determine usage policy based on risk assessment. Take into account individual and group roles within your organization. What policies are already in place? Do these users or groups already have access to the data used by Data Management Server? Are they Data Management Studio users? Generally, users will fall into one of the following categories: administrators, power or privileged users, general users, partners, and guests or external users. The approach to *deny all, allow as needed* will help you to implement security from the top down. New users should have restricted access. Access for administrators and power users could then be conferred manually or through explicit group authorizations.
>
> **Security Response** — Consider establishing a security response policy. If you have a security response team, specify how they are to respond to and report violations of security policy. Consider training all users on acceptable use prior to deployment of Data Management Server.

An important aspect of a security policy is to specify one or more groups of administrators.

Security Examples

There are two types of security available with Data Management Server, IP-based security, and DataFlux Authentication Server (Authentication Server). IP-based security, configured in the `dmserver.cfg` file, controls user access by IP address. For more information, see Control Access by IP Address . The Authentication Server is part of the DataFlux Data Management Platform. Through the Authentication Server client, user access can be controlled based on user, group, and job level authorizations. These security tools can be used separately or together. The following scenarios employ different types of security:

Scenario 1: Users in a small, local group use a specific range of IP addresses.

Scenario: Users have static IP addresses or draw dynamic addresses from a known range. If the group is small, or licenses are restricted to only a few machines, this may be the highest level of security needed by your organization.

Security plan: You can restrict access to Data Management Server by specifying IP addresses of clients that are allowed or denied access. Access can be restricted by general access, post/delete access, and restrictions on requests for statuses of jobs.

Scenario 2: Your organization requires control over user and group level access.

Scenario: Different users or groups require different levels of access, or certain files may require different authorizations.

Security plan: The Data Management Server security subsystem provides this degree of control. User name and password are passed using basic HTTP authentication to Data Management Server. Information on that user's user authorizations, group authorizations, and file authorizations are kept in Data Management Server security files. The Data Management Server security subsystem can be used alone or with IP-based security. The following is an example of basic HTTP authentication:

Client request:

```
GET /private/index.html HTTP/1.0
Host: localhost
```

Server response:

```
HTTP/1.0 401 UNAUTHORIZED
Server: HTTPd/1.0
Date: Sat, 27 Nov 2004 10:18:15 GMT
WWW-Authenticate: Basic realm="Secure Area"
Content-Type: text/html
Content-Length: 311
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN""http://www.w3.org/TR/1999/REC-html401-
19991224/loose.dtd">
<HTML>
 <HEAD>
 <TITLE>Error</TITLE>
 <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-
8859-1">
 </HEAD>
 <BODY><H1>401 Unauthorised.</H1></BODY>
</HTML>
```

Client request:

```
GET /private/index.html HTTP/1.0
Host: localhost
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

Server response:

```
HTTP/1.0 200 OK
Server: HTTPd/1.0
Date: Sat, 27 Nov 2004 10:19:07 GMT
Content-Type: text/html
Content-Length: 10476
```

Scenario 3: The Data Management Server Security Administrator wants to remotely administer a large number of users.

Scenario: The administrator wants to perform administrative tasks from the command line.

Security plan: Data Management Server security remote administration consists of SOAP commands to administer Data Management Server users and groups. This remote functionality allows the administrator to: change passwords; list all users; list all groups; list user's groups; list group's members; add user; set user's authorization; add group; delete account; add account to group; and delete account from group. Data Management Server must be running and security enabled.

Create a Group of Administrators

To create a group of users who will have administrative permissions on the Data Management Server, follow these steps:

1. Open ***install-path*\etc\dmserver.cfg**.
2. Specify the name of the group in the configuration option DMSERVER/SECURE/GRP_ADMIN.
3. Specify the names of the users in the group in the configuration option DMSERVER/SECURE/GRP_ALLOW. The user names must be defined on the Authentication Server.
4. Restart the Data Management Server to activate the configuration options.

Manage Permissions

Access Control Entry (ACE) for USERS and PUBLIC Groups

Set Permissions for Users and Groups

Set Permissions Using a Job List

Remove Users and Groups

Permissions Reference

Access Control Entry (ACE) for USERS and PUBLIC Groups

Two configuration options are available to set security for the default groups set by Authentication Server, PUBLIC and USERS. These options allow an administrator to specify the default object-level access for members of the USERS and PUBLIC groups respectively.

- DMSERVER/SECURE/DEFAULT_ACE_USERS = [0|1]
- DMSERVER/SECURE/DEFAULT_ACE_PUBLIC = [0|1]

Edit **dmserver.cfg** to set values for these configurations. The default setting for each of these configuration options is **0** which essentially blocks users without explicit permissions on each object. If set to **1** the default ACL created by Data Management Server will grant access rights to that object for that group set.

Consider the following when using these configuration options:

- This is only supported for the Authentication Server PUBLIC / USERS groups.
- The configuration only works for *new* objects with a default ACL set by the server. It does not work for any object with an existing ACL.
- Note that these permissions only apply to access rights on a specific object.
- Command permissions that are not object based, such as List/Post, are not affected by this setting.

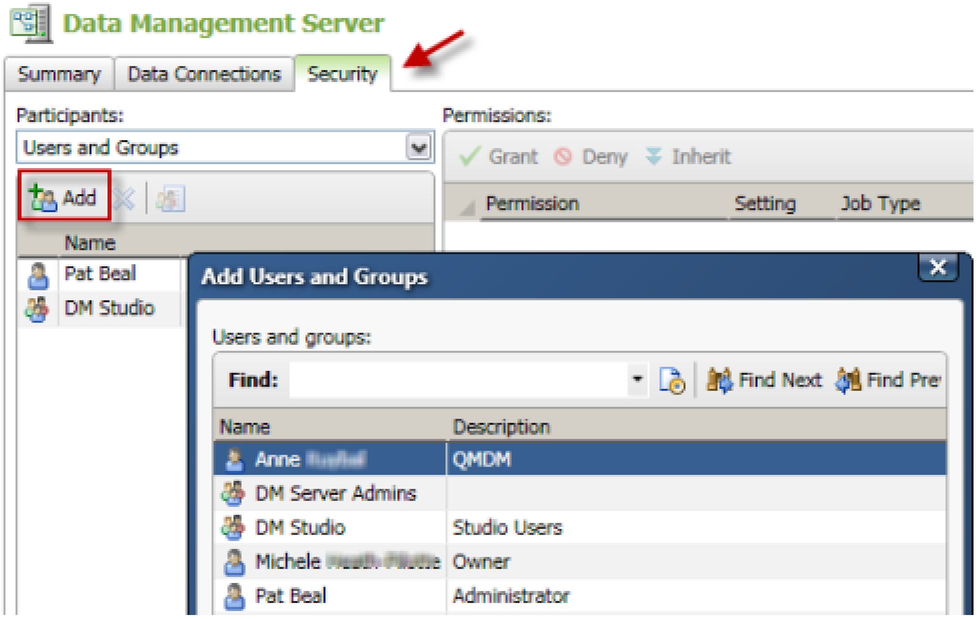
Following are possible scenarios in which using these configurations might be of benefit:

- A user creates an object, a job or a service, on the Data Management Server. When this is done, the default Access Control List (ACL) grants access rights to that object to that user only and makes them the owner of the object. If he needs to make that object available to other users or groups, he has to explicitly set whatever ACL he needs.
- A user posts numerous jobs and/or services to the Data Management Server and wants to allow other users to run these jobs. That user can use these options to avoid setting an ACL manually on each one of the posted objects.

Set Permissions for Users and Groups

Follow these steps to set permissions for users. To implement groups , you first create the group and define its members in theAuthentication Server. After this initial step is complete, the group is visible in the Data Management Server's **Add Users and Groups** dialog.

1. Log into Data Management Server and open the **Security** tab.
2. Select **Add** to open the *Add Users and Groups* dialog
3. Select a name from the list and click **Add**

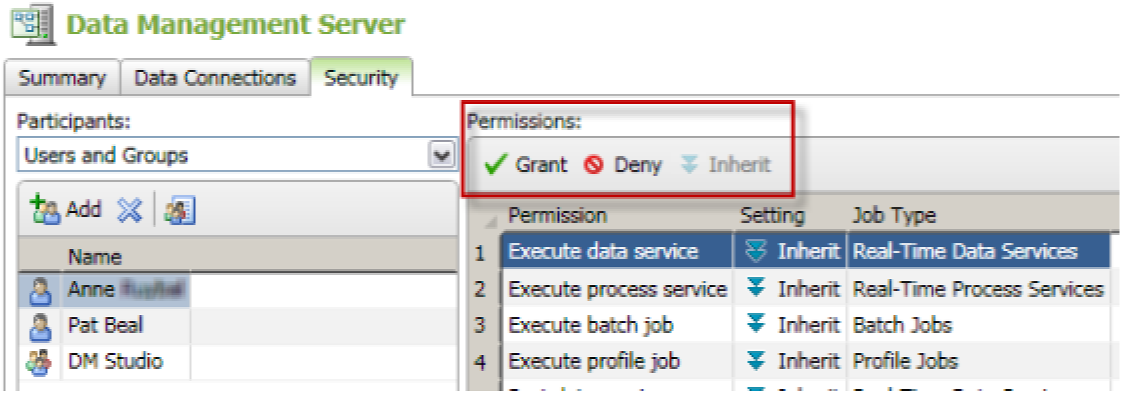


Name
Pat Beal
DM Studio

Name	Description
Anne [redacted]	QMDM
DM Server Admins	
DM Studio	Studio Users
Michele [redacted]	Owner
Pat Beal	Administrator

Data Management Server: Add Users and Groups Dialog

4. Click anywhere in the right information pane to activate the **Grant** and **Deny** Permissions icons. Assign permissions for the user you have just added by selecting a row at a time or holding down *shift + left mouse click* to select multiple rows at once.



	Name
👤	Anne [redacted]
👤	Pat Beal
👥	DM Studio

	Permission	Setting	Job Type
1	Execute data service	Inherit	Real-Time Data Services
2	Execute process service	Inherit	Real-Time Process Services
3	Execute batch job	Inherit	Batch Jobs
4	Execute profile job	Inherit	Profile Jobs

Data Management Server: Permissions List

Set Permissions Using a Job List

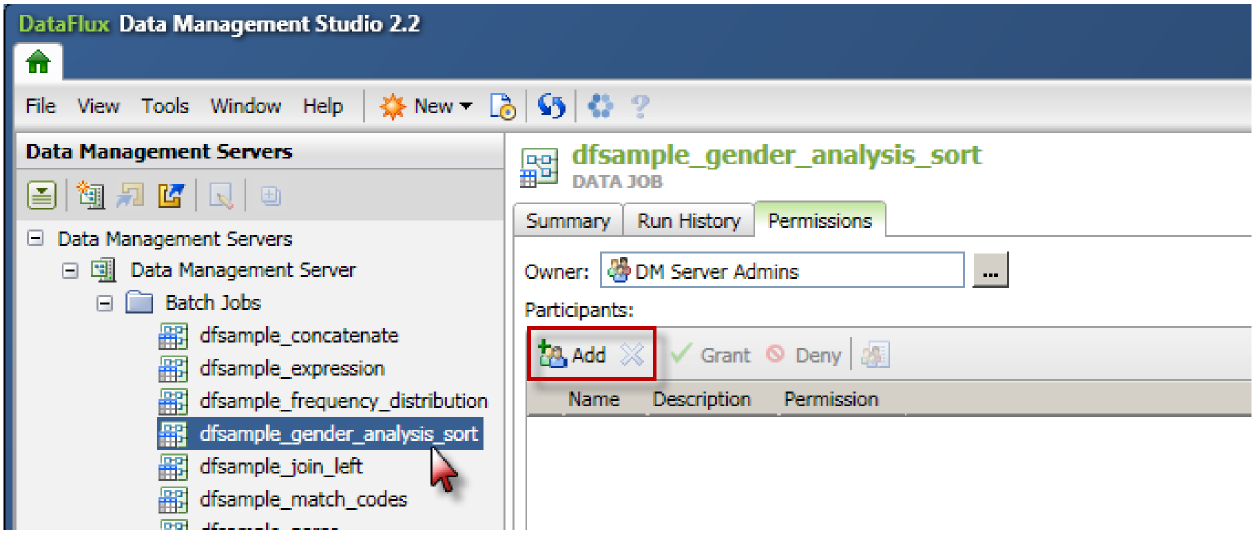
When a user posts an object (a job or service) to the server, that user is automatically set as the owner of the job, and the owner of an object will always be able to execute and delete an object, regardless of user or group authorizations. When a user creates an object by *copying* the file, ownership is set to the administrators group. An administrator can change ownership to another user or group at any time.

Use the following procedure to grant permissions directly from a job list in Data Management Server for Batch Jobs and Real-Time Services. Permissions for job objects are set at *Grant* or *Deny*.

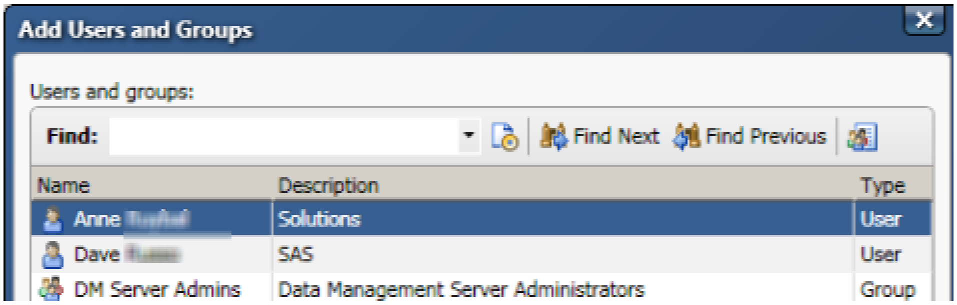
Note: Profile jobs do not have associated object-level access control, so you cannot set permissions for these job types.

1. Open Data Management Studio and click the **Data Management Servers** riser bar.
2. In the left navigation pane, select the Data Management Server you want to work with and log into that server.
3. Click the + sign next to your server to expand a list of job folders.
4. Click the + to expand the category of jobs or services you wish to work with: Batch Jobs, or Real-Time Data or Process Services. You cannot grant user permissions with profile jobs since these jobs do not have object-level access control.
5. Select a job from the Batch Job list in the left navigation pane, and then select the **Permissions** tab in the right information pane.
6. Under Participants, click **Add** to open the *Add Users and Groups* dialog.

Note: If the Permissions tab does not display, you might be viewing a Profile Job that does not have object-level access control. Permissions can be granted on Batch Jobs and Real-Time Services only.

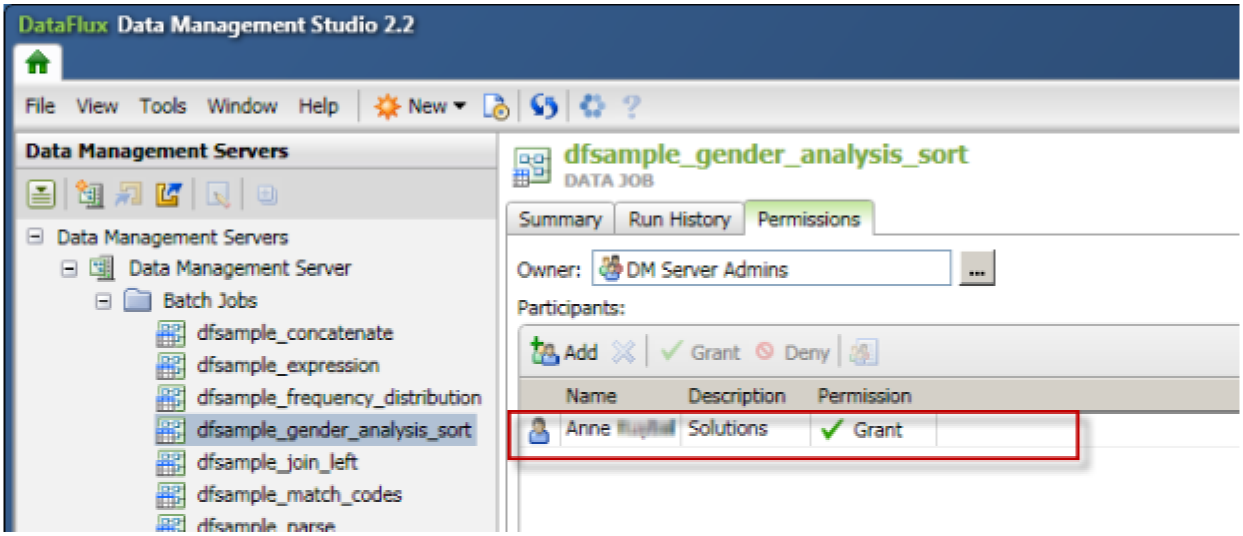


7. Select a user, or multiple users, and click **Add**.



Name	Description	Type
Anne [redacted]	Solutions	User
Dave [redacted]	SAS	User
DM Server Admins	Data Management Server Administrators	Group

8. The user is added to the participant list for the job and granted permissions.



Name	Description	Permission
Anne [redacted]	Solutions	✓ Grant

Note: At the Permissions tab, you can also change ownership of a job or service by clicking ... to the right of the **Owner**: field.

Remove Users and Groups

You can remove a user or group object from the **Users and Groups** list but that does not delete the object from Data Management Server since all user and group objects reside on the Authentication Server. To remove a user or group,

1. Log into Data Management Server and open the **Security** tab.
2. Select the user or group that you want to remove and click **delete**.
3. Click **Yes** at the confirmation dialog.

When the object is removed, its associated permissions are deleted.

Reference for Permissions

Permissions on the Data Management Server are defined as follows.

Permission	Description
Execute data service	When this option is enabled, the user can view and execute real-time data services. This includes run, preload, and unload a data service.
Execute process service	When this option is enabled, the user can view and execute real-time process services. This includes run, preload, and unload a process service.
Execute Batch Job	When enabled, the user can run a batch job, get a batch job file and get a batch job nodes' status.
Execute Profile Job	When enabled, the user can get and run a profile job.
Post Data Service	When enabled, the user can upload real-time data services to the server.
Post Process Service	When enabled, the user can upload real-time process services to the server.
Post Batch Job	When enabled, the user can upload a batch job to the server.
Post Profile Job	When enabled, the user can upload a profile job to the server.
Delete Data Service	When enabled, the user can delete a real-time data service.*
Delete process service	When enabled, the user can delete a real-time process service.*
Delete batch job	When enabled, the user can delete a batch job.*
Delete profile job	When enabled, the user can delete a profile job.*
List data service	When enabled, the user can list real-time data services.
List process	When enabled, the user can list real-time process services.

Permission	Description
service	
List batch job	When enabled, the user can list batch jobs.
List profile job	When enabled, the user can list profile jobs.

* In addition to enabling this permission, the user must also be the owner of the object, or an administrator, when performing these delete functions.

Control Access by IP Address

You can specify the following options to control access to the Data Management Server by IP address.

Setting	Description and Example
DMSERVER/IPACC/ALL_REQUESTS = allow *IP-list-or-range* | deny *IP-list-or-range* | allow all | allow none | deny all | deny none	Allows or denies the ability to connect to the Data Management Server. If this option is not specified, then the default value is **allow all**. For example: DMSERVER/IPACC/ALL_REQUESTS = allow 192.168.1.1-192.168.1.255
DMSERVER/IPACC/POST_DELETE = allow *IP-list-or-range* | deny *IP-list-or-range* | allow all | allow none | deny all | deny none	Allows or denies the ability to post and delete jobs. If this option is not specified, then the default is **allow all**. For example: DMSERVER/IPACC/POST_DELETE = 127.0.0.1
DMSERVER/IPACC/NOSECURITY = allow *IP-list-or-range* | deny *IP-list-or-range* | allow all | allow none | deny all | deny none	Allows or denies the ability to bypass all security checks on the Data Management Server. If this option is not specified, then the default value is **allow none** (no IP will bypass security checks). For example: DMSERVER/IPACC/NOSECURITY = allow 127.0.0.1 192.168.1.190 192.168.1.309

A list of IP addresses is formatted using blank spaces.

A range of IP addresses formatted with a dash character ('-') between the low and high ends of the range.

If any option entry contains all or none, then any specified IP addresses are ignored for that option.

Upgrade to SSL and AES

To maximize the security of your Data Management Server, install the DataFlux Secure software. DataFlux Secure enforces all SOAP connections to use the Secure Sockets Layer, with its HTTPS addresses. DataFlux Secure also upgrades encryption ,, on disk and on the network, from the default SASPROPRIETARY alrgorithm to the 256-bit AES algorithm. FIPS compliance ensures that the serverhat the server nsure secure communication between the server and its SOAP clients. To enable Data Management Server and other DataFlux products to work with SSL, installation of DataFlux Secure is required. Reference the *DataFlux Secure Administrator's Guide* for further information.

To enable SSL on the server, a few configuration changes are needed in the **dmserver.cfg** file. When enabling SSL, the following items can also be configured:

- The required key file used for authentication.
- The password for the key file. If this file is not password-protected, the KEY_PASSWD setting can be commented out.
- Certificates:
    - The Certificate Authority (CA) *file* (which stores trusted certificates) OR
    - a *path* to a directory that stores trusted certificates.

The following table contains the configuration settings for SSL.

Enabling SOAP with SSL

Edit the following settings as they apply to your environment. Configure these settings in the **dmserver.cfg** file located in install-path**\etc**.

Important: Stop the Data Management Server service or daemon before making any changes to the configuration file.

Configuration Option	Description
DMSERVER/SOAP/SSL	Set this value to **yes**. Once enabled, the following 4 settings should be configured if they apply.
DMSERVER/SOAP/SSL/KEY_FILE	Specifies the path to the key file that is required when the SOAP server must authenticate to clients.
DMSERVER/SOAP/SSL/KEY_PASSWD	Specifies the password for DMSERVER/SOAP/SSL/KEY_FILE. If the key file is not password protected, this configuration should be commented out.
DMSERVER/SOAP/SSL/CA_CERT_FILE	Specifies the file where the Certificates Authority stores trusted certificates.

Configuration Option	Description
DMSERVER/SOAP/SSL/CA_CERT_PATH	Specifies the path to the directory where trusted certificates are stored.

Configuring OpenSSL

There are numerous OpenSSL products on the market and many of them install differently; therefore, it is important to check the location of the program's DLL files after installation is complete.

If the installation package copies the DLLs into the System32 directory, that is the correct location for the files, so you do not need to do anything. In the event that the DLLs are *not* copied to System32, there are 3 options:

1. Manually copy the DLLs to System32,
2. Copy the DLLs to the install-path \bin directory of Data Management Server and other products.
3. The third choice, and possibly the easiest, is to add the bin directory of the OpenSSL install to the **PATH** environment variable.

Upgrade Path for Security-Related Objects

Data Management Server security-related settings and objects must be upgraded manually. Before migrating old command permissions and ACL file settings, you must configure the Authentication Server. If you used the Lightweight Directory Access Protocol (LDAP) with previous versions of the server user authentication, you can configure the Authentication Server to use the same LDAP server.

Manually upgrade the security-related objects in the following order:

1. Upgrade Users and Groups Files
2. Upgrade Jobs, ACL Settings, and Command Permissions

   Note: After completing this step, you should start the Data Management Server, and then start Data Management Studio (Studio). Next, add an instance of your Data Management Server to Studio, and log into that instance as the administrator.

Upgrade Users and Groups Files

Any users from the previous `users` file have to be added to the Authentication Server manually. Those users can then log into the Authentication Server to set their passwords. Any groups from the previous `groups` file must also be manually added to the Authentication Server. All group membership relationships from the **groups** file have to be manually recreated on the Authentication Server. For more information on creating users and groups on the Authentication Server, see the *DataFlux Authentication Server User's Guide*.

Upgrade Jobs, ACL Settings, and Command Permissions

Before upgrading ACL settings, the previous jobs should be copied into the Data Management Server. The architect jobs and services do not need to be converted to Data Management Platform-style jobs. All existing profile jobs must be imported into the repository.

Prior to upgrading existing user and group command permissions and ACLs, the Data Management Server must be running and configured with security enabled. It must also have a configured Authentication Server and have a Data Management Server administrators group set to a group that exists on the Authentication Server.

To upgrade existing user and group command permissions, look at the **users** and **groups** files (either by opening the files in a text editor or via the Authentication Server client) and configure the corresponding command permissions settings for the users and groups using Studio.

To upgrade existing ACL settings, look at the ACL files (either by opening the files in a text editor or via the Authentication Server client) and configure the ACLs for those same jobs using Studio. Alternatively, to configuring all ACL entries for files that used to be owned by specific users, you can create and configure the new ACL with that same user as the owner, and let those owners configure ACLs for their own jobs and services, using Studio.

For more information on Studio and configuring the Data Management Server, ACLs, and command permissions, see *DataFlux Data Management Studio Online Help*.

Managing Data Sources and Connections

Overview

Data Management Server allows you to create and maintain data source name (DSN) connections, which are also referred to as data connections. Data connections can be used to access ODBC, localized DSN, SAS data sets, and custom non-ODBC connections. You can create ODBC credentials for data sources that require login credentials with the **ODBC Credential Manager**. This allows you to make connections to that data source without being prompted for login credentials. When a job is run, the saved user credentials are retrieved and used. This allows the job to bypass saving credentials within the job and only references the data connection via the DSN name at runtime.

When running a deployed job, the Data Management Server may use the named connection and corresponding saved user credentials. Connections created on the Data Management Server must match the names of the client connection in Studio.

Another way to store user credentials is to associate a connection with a domain. Referred to as a **Domain Enabled ODBC Connection**, this feature can be used when an Authentication Server has been configured and the user logs into the Authentication Server. At this point, when accessing a data connection that has an associated domain, the login credentials are looked up via the Authentication Server and those credentials are used when the job is run.

Configure the Data Access Component (DAC)

The Data Access Component (DAC) allows the Data Management Server) to communicate with databases and manipulate data. The DAC uses Open Database Connectivity (ODBC) and Threaded Kernel Table Services (TKTS).

ODBC database source names (DSNs) are **not** managed by the DAC, but by the Microsoft ODBC Administrator. TKTS DSNs, however, are managed by the DAC and connections stored in the DSN directory.

The default DAC options should be satisfactory for most sites and should change only if your installation requires additional configuration. Most of the DAC settings come from configuration values as specified in **app.cfg**. It's safest to put DAC settings in the **app.cfg** file, however, they can be stored in the **macro.cfg** file also.

The following table lists the appropriate key that should be added to the **app.cfg** file and the default location where the file will be saved if the key and value pair is not specified:

Setting	Default Value
DAC/DSN	*install-path*\etc\dftkdsn\
DAC/SAVEDCONNSYSTEM	*install-path*\etc\dsn\

For more information on the **app.cfg** file, see the *DataFlux Data Management Studio Installation and Configuration Guide*.

For a complete list of Data Access Component options, see the *DataFlux Data Management Studio Online Help*.

Create Data Connections

- Overview
- Display Data Connections
- Create a Localized DSN Connection
- Create a Domain-Enabled ODBC Connection
- Create a Custom Connection
- Create a SAS Connection
- Edit a Data Connection
- Delete a Data Connection

Overview

The following data connections are available for configuration with Data Management Server:

- **Localized DSN Connection** - Enables you to create a localized Data Source Name (DSN) connection definition for a Federation Server and a specific data source. This connection definition is used when you access federated tables. These localized connections are Federated Server connections to a DBMS that are named and created as an extra connection to the same source in metadata.

- **Domain Enabled ODBC Connection** - Enables you to create a domain-enabled ODBC connection. A domain-enabled ODBC connection links a DataFlux Authentication Server domain to an ODBC Data Source Name (DSN) connection. The credentials for each user are stored on the Authentication Server and are automatically applied when the user accesses the domain-enabled connection. This approach provides more granular control over which credentials are used to access data.

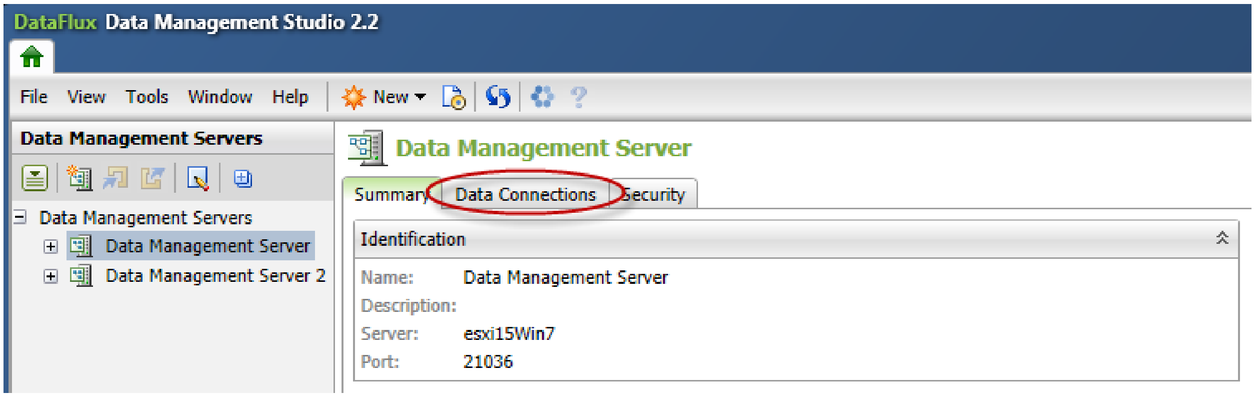
- **Custom Connection** - Enables you to create a custom connection string for non-ODBC connection types. These custom strings enable you to establish native connections from a DataFlux Federation Server to third-party databases or to draw data from more than one type of data input.

- **SAS Data Set Connection** - Enables you to create SAS data set connections.

Display Data Connections

Follow these steps to display the data connections that have been created for theData Management Server:

1. Open Data Management Studio and select the Data Management Server riser bar.
2. Select a registered Data Management Server and log in if prompted.
3. Select the **Data Connections** tab to open a dialog for creating data connections.



Name: Data Management Server
Description:
Server: esxi15Win7
Port: 21036

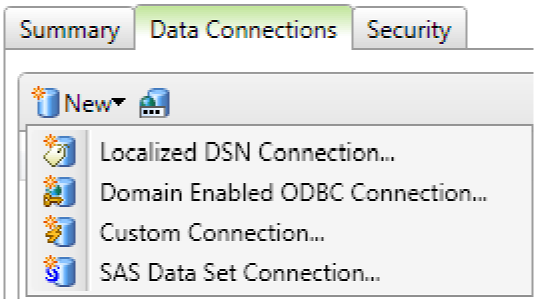
Create a Localized DSN Connection

You can use the Localized DSN dialog to create a connection to federated tables. Data federation provides a virtual way to use data from multiple sources without physically manipulating or moving data. Data can be left in the systems where it is created. The Federation Server makes federated queries for on-demand data access.

> **Important:** Connections to Data Integration Server version 8.2 and previous versions are not supported.

Follow these steps to create a localized DSN connection:

1. From Data Management Studio, click the **Data Management Servers** riser to access a list of Data Management Servers.
2. Select a server from the **Data Management Servers** list, and click the **Data Connections** tab.
3. To create a connection, select a connection type from the drop-down menu and enter the required information for the connection.


Summary
Data Connections
Security
New
Localized DSN Connection...
Domain Enabled ODBC Connection...
Custom Connection...
SAS Data Set Connection...

4. Click the **Data Connections** tab and select Localized DSN Connection from the drop-down menu.

5. Enter a name and description for the new connection into the appropriate fields.

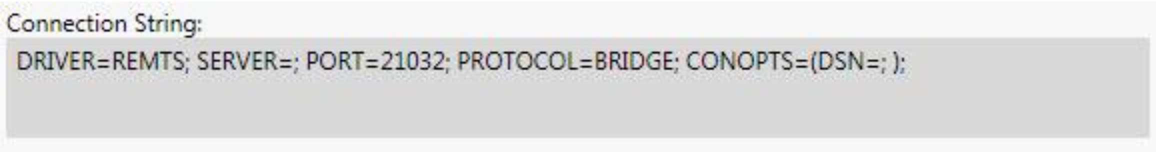
6. Enter the **Federated DSN**.

7. Enter the host name for the Federation Server in the Server field.

8. Enter the **Port** number.

9. Select a setting for credentials setting in the **Credentials** section of the dialog.

10. Note that the connection string created with the new connection is displayed in the Connection String field:



Connection String:

DRIVER=REMTS; SERVER=; PORT=21032; PROTOCOL=BRIDGE; CONOPTS=(DSN=;);

11. Click **OK** to save the new connection and return to the Data riser.

12. To verify a connection, double-click the connection in the **Data Connections** folder in the left pane. Enter any required credentials. If the connection works, you will be able to see tables, fields, and other attributes in the right panel

Create a Domain-Enabled ODBC Connection

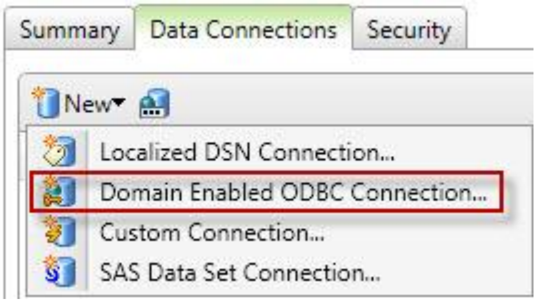
Domain-enabled connections are created in Data Management Studio and require an Authentication Server. To execute jobs on a Data Management Server, the DSN connection that is referenced in the domain-enabled ODBC connection must be defined in a location that is accessible to the Data Management Server. Otherwise, the reference to the DSN in the domain-enabled ODBC connection will not resolve. After the connection is created in Data Management Studio, copy the **.dtfk** file to the Data Management Server in ***install-path*****\etc\dftkdsn**.

You can also create the domain enabled connection in Data Management Server using the following procedure:

1. Open a registered Data Management Server using the riser bar in Studio and log in if prompted.

2. Click the **Data Connections** tab and click **New**.

3. Select **Domain Enabled ODBC Connection** from the drop-down menu.



Summary
Data Connections
Security
New
Localized DSN Connection...
Domain Enabled ODBC Connection...
Custom Connection...
SAS Data Set Connection...

4. In the **Name** field, enter a name for the connection, preferably referring to the Domain that was created in Authentication Server.
5. Enter some information about the domain in the **Description** field.
6. Select the ODBC DSN that was configured in the Windows ODBC Administrator.
7. In the Domain name field, enter the actual domain name as it was created in Authentication Server
8. Click **OK**

The new connection places a **.dftk** file in the ***install-path*\etc\dftkdsn** directory.

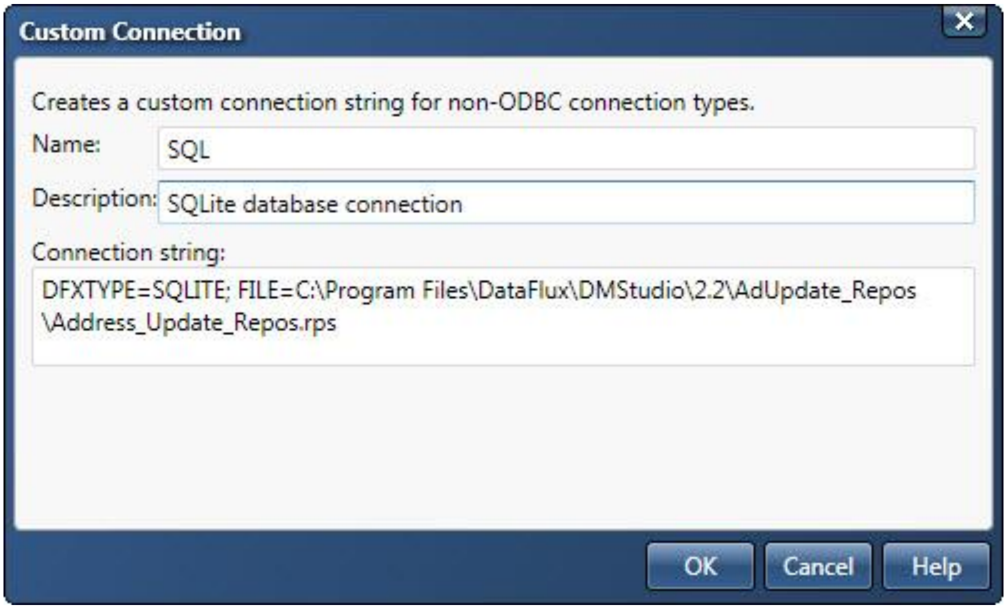
Create a Custom Data Connection

Custom connections enable you to access data sources that are not otherwise supported in the Data Connections interface of Data Management Server. Examples of custom connections are those that connect to SAP or SQLite.

Follow these steps to connection:

1. In Studio, connect to the Data Management Server using the riser bar and log in if prompted.
2. Click the **Data Connections** tab and click **New**.
3. Enter the connection string into the Custom connection field. Specify values for the attributes that are shown in the example connection string. The value for FILE must be the complete path to the RPS file.

```
DFXTYPE=SQLITE;
FILE=C:\Program Files\DataFlux\DMStudio\2.2\AdUpdate_Repos\Address_Update_Repos.rps
```



Creates a custom connection string for non-ODBC connection types.

Name: SQL

Description: SQLite database connection

Connection string:

DFXTYPE=SQLITE; FILE=C:\Program Files\DataFlux\DMStudio\2.2\AdUpdate_Repos\Address_Update_Repos.rps

4. Click **OK** to save the new connection.

Create a SAS Connection

Follow these steps to create a connection to a SAS data set:

1. Click the **Data Connections** tab and click **New**.
2. Select **SAS Data Set Connection** from the drop-down menu.
3. Enter a name and description for the new connection into the appropriate fields.
4. Enter the path to the directory that contains the SAS data set that you want to connect to.
5. Verify that the appropriate option is specified in the **Access** field. DEFAULT is read-write access. READ-ONLY assigns a read-only access. TEMP specifies that the data set be treated as a scratch file.
6. Verify that the appropriate option is specified in the **Compression** field. If you compress data, you might experience a slowdown in performance. NO specifies that the observations are uncompressed. Use YES or Character to compress character data. Use BINARY to compress binary data.
7. Verify that the appropriate option is specified in the **Table Locking** field. SHARE specifies that other users or processes can read data from the table but prevents other users from updating. EXCLUSIVE locks tables exclusively which prevents other users from accessing any table you open.
8. Verify that the appropriate option is specified in the **Encoding** field. The default is the SAS System encoding. You might select an encoding that is different than the default if you are processing a file with a different encoding.

   Note: You can select an encoding from the drop-down menu. If you type the first few letters of the desired encoding, the closest match will be added to this field. For more information, see the appendix, Encodings for SAS Data Sets.

9. **Connection String** - Displays the connection string for the SAS data set. Check the connection string to see if the appropriate options encoding has been selected for this connection. You can test the connection by clicking Test Connection.
10. Click **OK** to save the new connection.

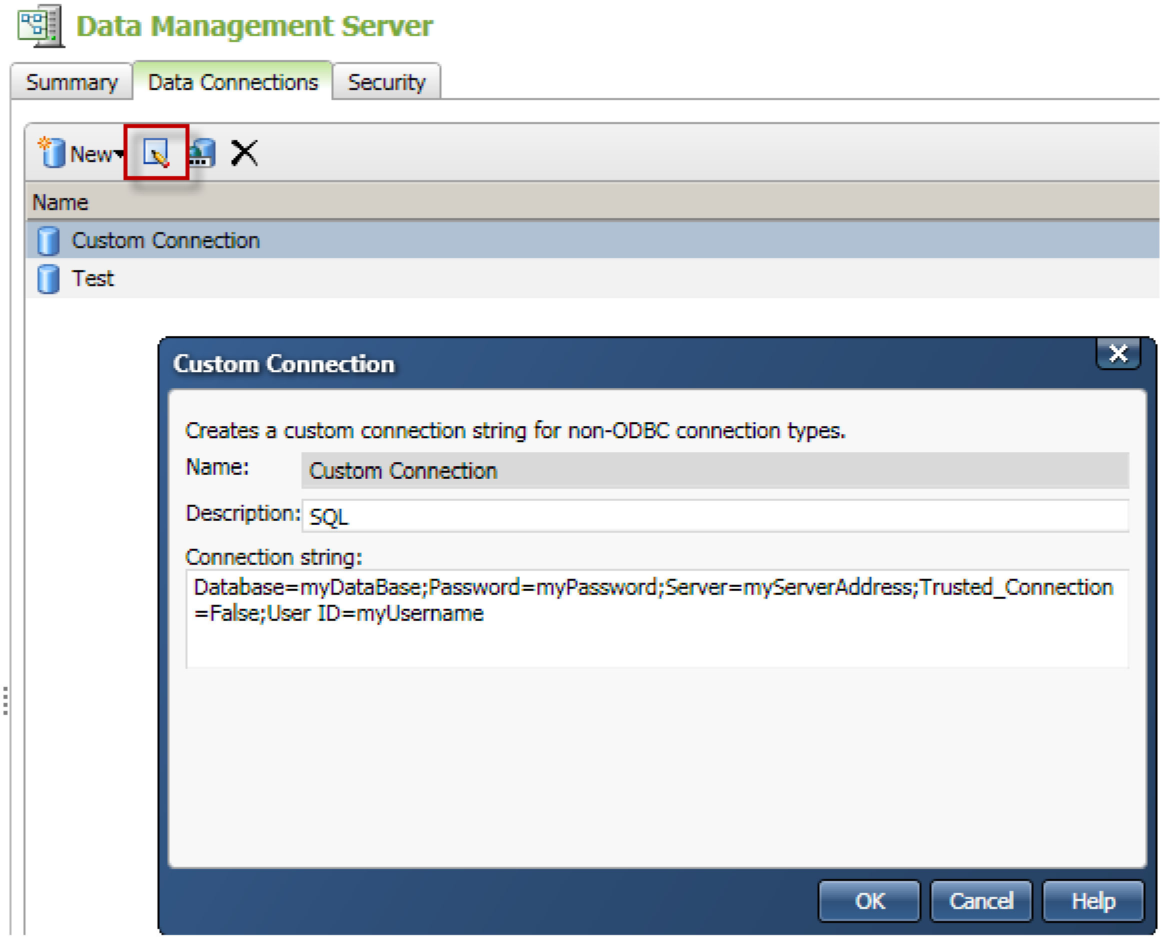
Edit a Data Connection

Follow these steps to edit a data connection:

1. From Data Management Studio, click the **Data Management Servers** riser to access a list of Data Management Servers.
2. Select the name of the server for which you want to manage connections. If you are prompted, log on to the server by entering your user ID and password, and then click **Log On**.
3. In the information pane, click the **Data Connections** tab that presents a list of current connections.

4. Select a data connection and click **Edit**. The connection type will determine which fields and options are available for you to edit.

5. In the dialog that opens, make the desired changes and click **OK**.



Creates a custom connection string for non-ODBC connection types.

Name: Custom Connection

Description: SQL

Connection string:

```
Database=myDataBase;Password=myPassword;Server=myServerAddress;Trusted_Connection
=False;User ID=myUsername
```

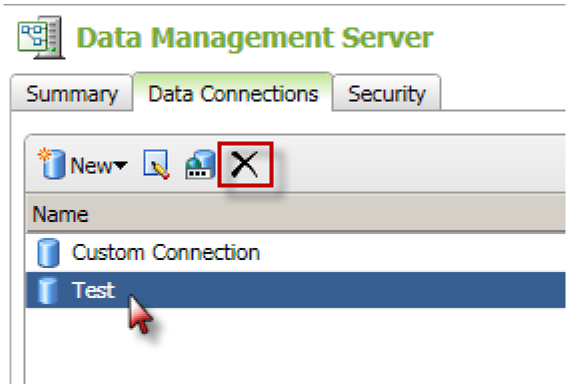
Delete a Data Connection

Follow these steps to delete a data connection:

1. From Data Management Studio, click the **Data Management Servers** riser to access a list of your Data Management Servers.

2. Click the name of the server for which you want to manage connections. If you are prompted, log on to the server by entering your user ID and password, and then click **Log On**.

3. In the information pane, under the **Data Connections** tab, select the name of the data connection that you want to delete and click **Delete**.

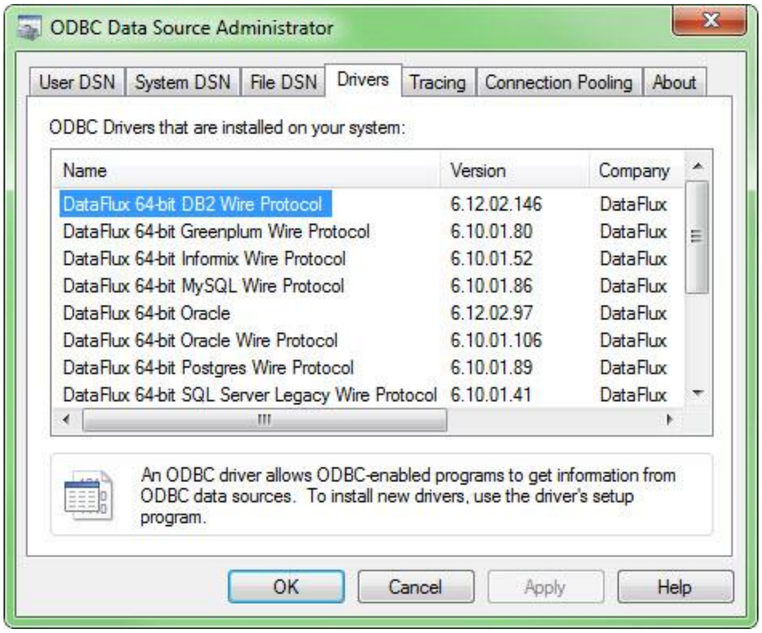


4. When prompted, confirm that you want to delete the data connection by clicking **Yes**.

Configure ODBC Connections

To process a database with Data Management Server, a DataFlux Driver for ODBC for the specified database must be installed, and the database must be configured as an ODBC data source. You can also access flat files and text files outside of the ODBC configuration method if your batch or profile job has specific nodes for those data sources.

DataDirect provides a number of wire protocol ODBC drivers that communicate directly with a database server, without having to communicate through a client library. These drivers are available with the Data Management Server installation files. If they have been installed, access them from the **Drivers** tab of the ODBC Data Source Administrator dialog.



ODBC Drivers that are installed on your system:

Name	Version	Company
DataFlux 64-bit DB2 Wire Protocol	6.12.02.146	DataFlux
DataFlux 64-bit Greenplum Wire Protocol	6.10.01.80	DataFlux
DataFlux 64-bit Informix Wire Protocol	6.10.01.52	DataFlux
DataFlux 64-bit MySQL Wire Protocol	6.10.01.86	DataFlux
DataFlux 64-bit Oracle	6.12.02.97	DataFlux
DataFlux 64-bit Oracle Wire Protocol	6.10.01.106	DataFlux
DataFlux 64-bit Postgres Wire Protocol	6.10.01.89	DataFlux
DataFlux 64-bit SQL Server Legacy Wire Protocol	6.10.01.41	DataFlux

An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.

DataFlux ODBC Drivers

When configuring a new data source for the Data Management Server, it is essential that the parameters match those used to create the data source on the client machine where Data Management Studio is installed.

If Data Management Studio and Data Management Server are installed and running on the same machine, you will need to set up the ODBC DSN two times; once through **ODBC Connections** in Data Management Studio and a second connection for Data Management Server using the Windows ODBC Data Source Administrator.

Adding ODBC Connections

ODBC connections for jobs run within Data Management Studio are created in Data Management Studio; however, ODBC connections for Data Management Server must be created on the server. Use the **Windows ODBC Data Source Administrator** for Windows servers, or the **dfdbconf** tool for server UNIX servers.

Windows ODBC Data Source Administrator

The Microsoft® ODBC Data Source Administrator manages database drivers and data sources. This application is located in the Windows Control Panel under Administrative Tools. Beginning in Windows 8, the icon is named **ODBC Data Sources**, and on 64-bit operating systems there is a 32-bit and 64-bit version.

To open the ODBC Administrator from the Control Panel,

1. Click **Start** and point to **Settings**.
2. Click **Control Panel.**
3. Double-click **Data Sources** (**ODBC**) to open the **ODBC Data Source Administrator** dialog box.
4. Select the **System DSN** tab and click **Add...**
5. Select the appropriate driver from the list and click **Finish**.
6. Enter your information in the **Driver Setup** dialog and click **OK** when finished.

UNIX ODBC Configuration Tool

The interactive ODBC Configuration Tool, **dfdbconf**, can be used to add new data sources to your ODBC configuration. From the root directory of the Data Management Server installation, run:

```
./bin/dfdbconf
```

Select a driver from the list of available drivers and set the appropriate parameters for that driver. The new data source is added to the **ODBC INI** file. You can also use **dfdbconf** to delete the data source if it is no longer needed.

Testing the Connection

Once you have added all your data sources, use the interactive ODBC Viewer, **dfdbview** to test your connection. For example, if you added a data source called *my_oracle*, run the following command from the installation root:

```
./bin/dfdbview my_oracle
```

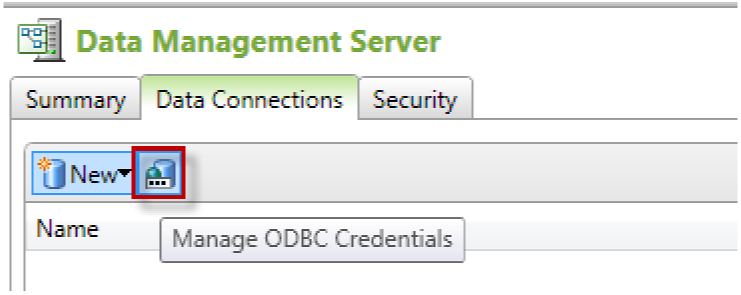
You may be prompted for a user ID and password if the connection is secured. If the connection succeeds, a prompt appears from where you can enter SQL commands and query the database. If the connection fails, error messages describing the reasons for the failure will be displayed.

Managing ODBC Credentials

An ODBC credential allows you to store login credentials once, in order to easily connect to a data source in the future. The credentials contain connection information so it does not need to be stored inside the job, or entered at the time the job is run. This allows for better protection and management of security, confidentiality, and a more versatile way to handle access to data sources that require authentication.

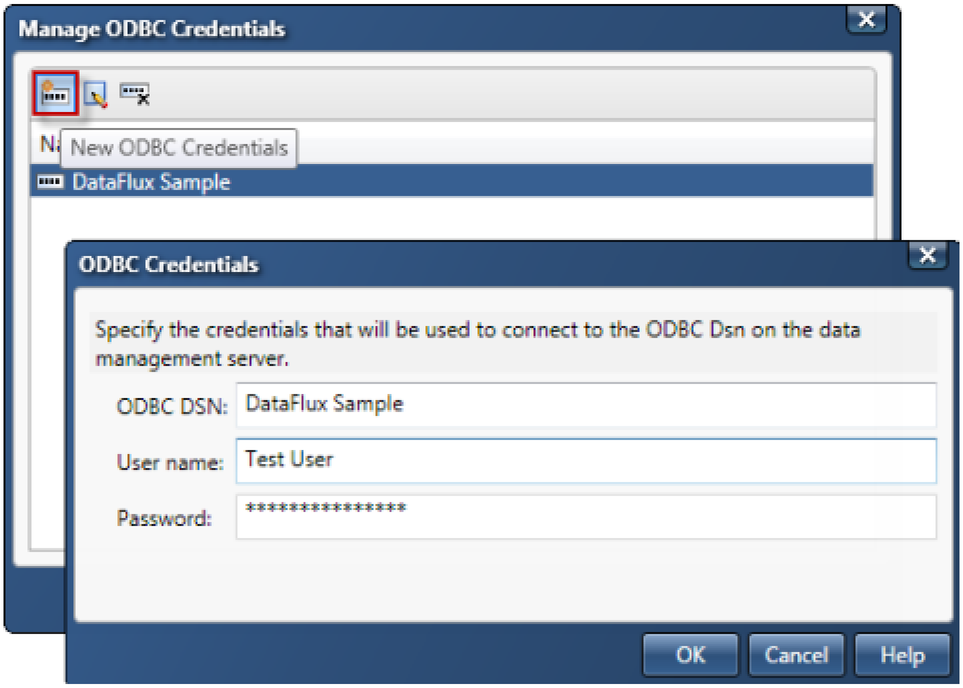
To manage ODBC credentials, complete the following steps:

1. From Data Management Studio, click the **Data Management Servers** riser bar to access a list of your Data Management Server(s).
2. Select the name of the Data Management Server for which you want to manage connections. If you are prompted, log on to the server by entering your user ID and password, and click **Log On**.
3. In the information pane, select the **Data Connections** tab and click the **Manage ODBC Credentials** icon.



Manage ODBC Credentials

4. Using the **Manage ODBC Credentials** dialog, you can create, edit and delete ODBC Credentials as outlined in the following items.
   - Create ODBC Credentials: Click **New ODBC Credentials** to create a new data source credential. Enter the ODBC DSN, user name and password, and click **OK**.



Specify the credentials that will be used to connect to the ODBC Dsn on the data management server.

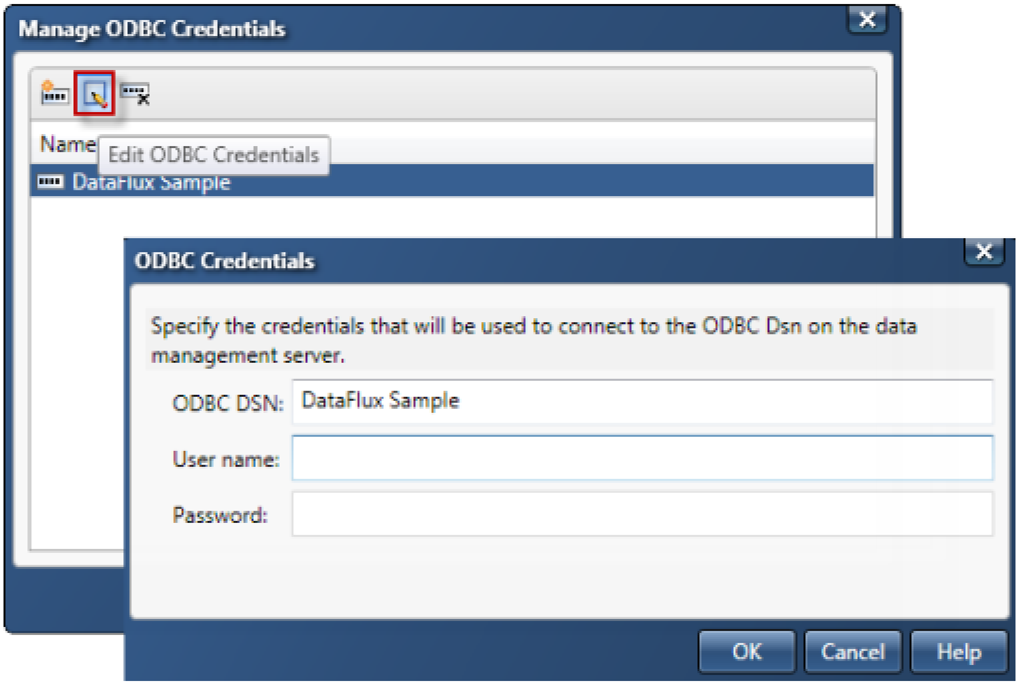
ODBC DSN: DataFlux Sample

User name: Test User

Password: ***************

New ODBC Credentials

- Edit ODBC Credentials: To edit an existing credential, select a name from the list and click the **Edit ODBC Credentials** icon. The **ODBC Credentials** dialog opens where you can make changes to the user name or password that will access the ODBC DSN. Click **OK** to close the ODBC Credentials dialog. If there are no saved credentials, the Edit ODBC Credentials icon is not active.



Specify the credentials that will be used to connect to the ODBC Dsn on the data management server.

ODBC DSN: DataFlux Sample

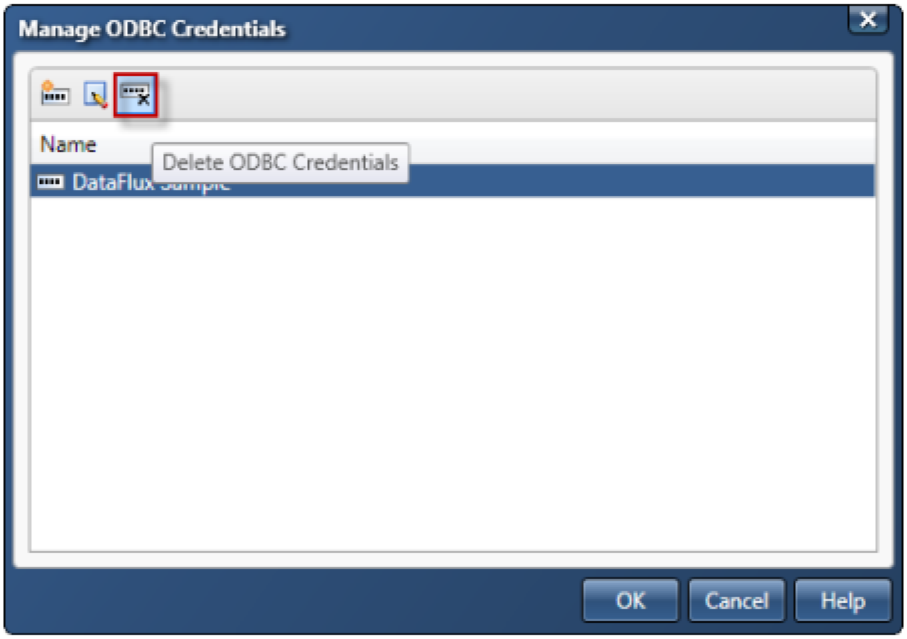
User name:

Password:

Edit ODBC Credentials

- Delete ODBC Credentials: Select a credential name and click **Delete ODBC Credentials** to remove the item. You can use **Ctrl** + **left click** to select more than one name. Click **OK** to close the Manage ODBC Credentials dialog when you are finished.

⚠**Warning**: Use caution when deleting an ODBC credential. When a name is deleted from the list, clicking Cancel will not reverse the deletion.



Delete ODBC Credentials

Managing Clients, Jobs, and Services

Configure the SOAP and WLP Servers

The Data Management Server manages client connections using multiple threads. By default, a SOAP server communicates with clients. To enhance performance, you can enable a second server that listens for client connections at a separate port. The two servers run as separate threads in a single process. Both servers spawn new threads to connect to clients, using a shared thread pool. The two servers are defined as follows:

- **SOAP server** — uses a SOAP interface, as defined in the DataFlux Web Service Definition Language (WSDL) file. For more information on the WSDL file, see the Code Examples appendix.
- **Wire-Level Protocol (WLP) server** — uses a proprietary WLP client library. WLP offers a significant performance increase over SOAP, especially for real-time services. The WLP server is disabled by default.

The Data Management Server processes a single SOAP request per client connection. After the server returns a response, the connection is closed. This is true even if the client attempts to make a persistent connection by including "**Connection: Keep-Alive**" (for HTTP 1.0) or omitting "**Connection: close**" (for HTTP 1.1) in the HTTP header of the SOAP request.

To manage the configuration of the SOAP and WLP server, set the following configuration options. For a complete list of configuration options, refer to the Configuration Options Reference for dmserver.cfg.

Configuration Option	Description
DMSERVER/SOAP/LISTEN_PORT	Specifies the port on which the SOAP server listens for connections. The default port is 21036.
DMSERVER/SOAP/LISTEN_HOST	Specifies the hostname or IP address to which the SOAP server must bind. A machine running Data Management Server might be available on the

Configuration Option	Description
	network under different hostnames or IP addresses, and different machines on the network may have access to this machine via one hostname or IP address, but not via others. By binding to a particular hostname or IP address, the server only hears requests addressed specifically to that hostname or IP address. A pair of hostnames and IP addresses can be used interchangeably. For example, if this option is set to **localhost**, local clients sending requests to 127.0.0.1 will still be heard. However, requests to public IP address or the hostname of the machine (or requests coming from external clients to the machine) will not be heard. By default, this option is left blank. That means Data Management Server is not bound to any specific hostname or the IP address of the machine and will receive all requests that are coming to the machine, on the correct port.
DMSERVER/WLP	Specifies if WLP server is enabled. Valid configuration options are **yes** and **no**. The default is **no**. If set to **yes**, WLP Server is started and uses its own listen port. The default is no meaning that WLP server is bypassed during startup of Data Management Server. This means that WLP clients can not connect to Data Management Server but SOAP clients can. The Data Management Server log will contain entries for the status of WLP server.
DMSERVER/WLP/LISTEN_PORT	Specifies the port on which the WLP server listens for connections from WLP clients. If you are running multiple instances of the server on the same machine, each instance must have a unique port configured for it. The default port is 21037.
DMSERVER/WLP/LISTEN_HOST	Specifies the hostname or IP address to which the WLP server must bind. By default, this option is left blank. For more information on binding to a hostname or IP address, see DMSERVER/SOAP/LISTEN_HOST.

Configure the Server to Pre-load Services

The following sections describe how to use pre-load configuration settings in your DataFlux® Data Management Server (Data Management Server) environment. Data Management Server can preload selected services on startup. This is helpful if you typically use the same services each time you run Data Management Server and would like to have these services available as soon as Data Management Server is running.

Several configuration options are available to preload services on the Data Management Server:

1. DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL = *count*
2. DMSERVER/SOAP/DATA_SVC/PRELOAD = *count*:*name_of_service* *count*:*name_of_service* ...

   ...where *count* specifies the number of preload instances and *name_of_service* indicates the name of the service element. This may include the directory where the service is located.
3. DMSERVER/SOAP/DATA_SVC/PRELOAD_DURING_RUN = *yes*|*no*

   By default Data Management Server preloads all configured services before starting to accept SOAP requests. The same applies if this option is set to ***no***. When this option is set to ***yes***, Data Management Server starts a separate thread to preload all configured services at run-time, while accepting SOAP requests at the same time. If Data Management Server is stopped while the preload thread is still running, that thread will be terminated.

Preload All Services

The first directive, **DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL** = *count*, causes Data Management Server to find and preload all services a specific number (*count*) of times. This includes services found in subdirectories. The number of instances of each service (*count*) must be an integer greater than 0, or the directive is ignored.

> *For example, DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL = 2 causes Data Management Server to preload two instances of each service that is available, including those found in subdirectories.*

Preload One or More Specific Services

The second directive, **DMSERVER/SOAP/DATA_SVC/PRELOAD** = *count*:*name_of_service*, designates the specific services, as well as the count for each service, that Data Management Server is to preload on startup. Use additional count and service elements, *count*:*name_of_service*, for each service. Separate each count and service element by one or more white space characters. The service element, however, cannot include white space characters. Additionally, all elements must be listed on a single line. Using this format, you can configure a directive that starts a number of services, with each service having a different count.

> *For example, DMSERVER/SOAP/DATA_SVC/PRELOAD = 2:abc.ddf 1:subdir1\xyz.ddf loads two counts of abc service, and one count of xyz service, located in the subdir2 subdirectory.*

Configure Complex Preloads

By combining the two options, you can configure more complex preloads. The two options add the counts arithmetically to determine how many services are actually loaded. Internally, Data Management Server builds a list of all services it needs to preload and, for each service, sets the total count.

The following two example options illustrate the logic of how this works:

DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL = 2

DMSERVER/SOAP/DATA_SVC/PRELOAD = 2:svc1.ddf -1:subdir1\svc2.ddf -2:svc3.ddf

The first directive instructs Data Management Server to preload a total of two instances of all existing services. The second directive modifies this in the following ways:

- Two additional counts of **svc1.ddf** are added, for a total of four instances. The counts are added together, and the total is the number of instances that Data Management Server tries to preload.
- The **svc2.ddf** file, which is found in the subdir1 `subdir1` subdirectory, has a -1 count. This produces a total count of one for `svc2.ddf`.
- For the `svc3.ddf` file, there is a combined total count of zero, so this service is not loaded at all. The value of *count* must be greater than zero for a service to be preloaded.

Some important points to remember:

- Data Management Server attempts to preload a single instance of all requested services before trying to preload more instances, if more than one instance is specified.
- The service element can include the path to the service, relative to the root of the services directory. For example, `1:subdir1\svc2.ddf` specifies one instance of service `svc2.ddf`, which is located in the `subdir1` subdirectory.
- Count can be a negative value. This is meaningful only when both configuration options are used together.
- Preloading stops when Data Management Server has attempted to preload all required instances (successfully or not), or if the limit on the number of services has been reached. Depending on whether a SOAP or WLP server is used, the limit can be specified by using one of the following configuration options: **DMSERVER/SOAP/DATA_SVC/MAX_NUM =**, or **DMSERVER/WLP/DATA_SVC/MAX_NUM =**. These configurations will default to 10 if a number is not specified.

Browse Available Services and WSDLs

You can use a Web browser to display lists of available data services and process services. You can also display definitions of available data services and process services. The definition files are formatted in the Web Service Definition Language.

You can configure your Data Management Server to generate WSDL service definitions dynamically, in response to GET WSDL requests.

To use a Web browser to display a list of available **data services**, enter an address in the following format:

http://*server-hostname*:*port*/datasvc/

As shown in this example:

http://dev083:21036/datasvc/

To use a Web browser to display a list of available **process services**, enter an address in the following format:

http://*server-hostname*:*port*/procsvc/

As shown in this example:

http://dev083:21036/procsvc/

To use a Web browser to **display the WSDL of a data service**, enter an address in the following format:

http://*server-hostname*:*port*/dataSvc/*path*/*service-name*?wsdl

The path is the directory path in *install-path*/share/web/data-services.

The following example displays the WSDL of the data service named RAM.DDF:

http://dev083:21036/dataSvc/proj1/memory/RAM.DDF?wsdl

To use a Web browser to **display the WSDL of a process service**, enter an address in the following format:

http://*server-hostname*:*port*/procSvc/*service-name*?wsdl

The path is the directory path in *install-path*/share/web/data-services.

The following example displays the WSDL of a process service named RAM.DDF:

http://dev083:21036/procSvc/RAM.DDF?wsdl

If a WSDL does not already exist for a data service or a process service, then one of the two things will happen. If the Data Management Server is configured to generate a WSDL in response to GET WSDL requests, then the server generates a WSDL for display in the browser. Otherwise, the browser displays an error.

To generate WSDLs in response to GET WSDL requests, set the following option in dmserver.cfg: DMSERVER/SOAP/WSDL/GEN_ON_GET = yes.

Apply SOAP Commands and WSDL Options

- SOAP Commands Reference
- Response from the GenerateWSDL Command
- Reference for Job-Specific WSDL Configuration Options

The Data Management Server supports a number of SOAP commands intended for running jobs and services, as well as administering certain aspects of the server and its security. Simple Object Access Protocol (SOAP) commands consist of request-and-response pairs that use simple types (integers and strings) or types (structures built from simple types and other structures). Definitions of all requests, responses, and complex types are found in the Web Service Definition Language (WSDL) file, which is in Data Management Server installation directory, in the **share** subdirectory.

Note: WSDL 2.0 is not supported.

SOAP Commands Reference

The following table lists the names of the SOAP commands that correspond to requests and responses in the WSDL file, as well as a description for each command:

Command	Description	Command Type
GenerateWSDL	Allows you to generate a WSDL for a single data job or process service job, multiple jobs, or for an entire directory including all associated subdirectories. You can pass job names or a single directory name but both cannot be used together in the command. Using a job and directory name together will result in an error. Data Management Server can generate multiple WSDLs within a request and will not stop if an error occurs. See Response from the GenerateWSDL Command to review the responses that result from the use of this command.	
GetServerVersion	Allows you to retrieve the server's version information, as well as versions of installed reference data, the repository, and some of the libraries. GetServerInfo also returns date and time elements.	Single version command
ArchitectServiceParam	Allows you to retrieve input and output fields of a data service.	Data services command
ArchitectService	Allows you to run a data service. ArchitectService has a *timeout* integer element to specify the number of seconds to let a data service run before stopping it. If the *timeout* element is omitted from request, or is set to 0, no timeout will occur. **Note:** The timeout count may be off by approximately a second or two due to rounding up of counts less than a second (e.g. 1.5 = 2 secs).	Data services command
ArchitectServicePreload	Allows you to start the requested number of data service processes and load specified data service jobs into those processes.	Data services command
ArchitectServiceUnload	Allows you to terminate the specified data service process.	Data services command
LoadedObjectList	Allows you to retrieve a list of running data	Data

Command	Description	Command Type
	service processes, along with the name of the loaded service job for each process.	services command
MaxNumJobs	Allows you to set the maximum number of concurrent data service processes. This is a run-time setting only and has no effect on the value in `dmserver.cfg` file.	Data services command
WorkFlowJobParams	Allows you to retrieve inputs and outputs of either a process service or a batch job.	Process services command
WorkFlowService	Allows you to run a process service.	Process services command
UnloadProcesses	Allows you to kill all idle dfwfproc processes and subsequent busy dfwfproc processes once they become idle.	Process services command
RunArchitectJob	Allows you to run a batch job.	Batch and profile jobs commands
RunProfileJob	Allows you to run a profile job.	Batch and profile jobs commands
TerminateJob	Allows you to terminate a running batch or profile job. The client can still get the status, log, and statistics file (if one exists) after the job has been terminated.	Batch and profile jobs commands
JobStatus	Allows you to retrieve status information for one or more batch or profile jobs. Applies to jobs that are running or already finished.	Batch and profile jobs commands
JobNodesStatus	Allows you to retrieve status information for every node in a batch job. Applies only to the jobs that are currently running.	Batch and profile jobs commands
JobLog	Allows you to retrieve the log file and statistics file (if one exists) for a batch or profile job. Applies only to already finished jobs.	Batch and profile jobs commands
DeleteJobLog	Allows you to delete a job log and statistics file (if one exists). This command removes all history for a given job run. Applies only to already finished jobs.	Batch and profile jobs commands
ObjectList	Allows you to retrieve a list of available objects	Object files

Command	Description	Command Type
	of a specified type (data services, process services, batch jobs, or profile jobs).	commands
PostObject	Allows you to upload an object of a specified type. If an object of that type with the same name and path already exists, an error is returned.	Object files commands
ObjFile	Allows you to download an object of a specified type.	Object files commands
DeleteObject	Allows you to delete an existing object of a particular type from the server.	Object files commands
ListAccounts	Allows you to retrieve a list of user and group IDs with explicitly configured server commands permissions (which are included).	Security commands
SetPermissions	Allows you to set server command permissions for an user or group ID.	Security commands
DeleteAccount	Allows you to delete server command permissions for an user or group ID.	Security commands
SetACL	Allows you to set an access control list (ACL) for an object (data service, process service, or batch job).	Security commands
GetACL	Allows you to retrieve an ACL for an object.	Security commands

Response from the GenerateWSDL Command

The response from a **GenerateWSDL** command contains two lists:

1. One list contains the job names for which WSDLs were generated successfully, and
2. The second list contains the job names for which WSDLs could not be generated. Each entry includes a detailed error message as to the cause of the problem.

Generating a WSDL is similar to obtaining properties for a service, therefore, similar errors are shown: '*job file not found*', '*access denied*', '*job format is invalid*', '*failed to connect to DB*...', etc.). The response is sent only when request is completed, with or without errors. The amount of time it takes to generate a WSDL is determined by the nature of the job and its dependencies.

Reference for Job-Specific WSDL Configuration Options

The table below reflects the available configuration options for job-specific WSDL and are located in **dmserver.cfg**. To support generation of job-specific WSDL from a Data Management Server in Data Management Studio, the following configurations, also listed in the table, are required:

DMSERVER/SOAP/WSDL=YES

DMSERVER/SOAP/WSDL/GEN=SINGLE | MULTIPLE

DMSERVER/SOAP/WSDL/GEN_ON_GET=YES

DMSERVER/SOAP/WSDL/RUN_IGNORE_MTIME=YES

Configuration Option	Description
DMSERVER/SOAP/WSDL	Specifies whether or not to load WSDLs when starting Data Management Server. If job-specific WSDL functionality is needed, set this option to **yes.** This will enable Data Management Server to load existing WSDLs on startup and to recognize jobs that are WSDL-based runSVC requests, that is, if matching WSDLs exist. This will also allow Data Management Server to recognize other WSDL configuration options. The default setting is **no**, which means that existing WSDLs are not loaded, new ones are not generated, no job WSDL-based runSVC requests are recognized, and other WSDL configuration options will not be processed.
DMSERVER/SOAP/WSDL/GEN	Specifies whether or not to allow generation of run-time WSDLs. The default setting is **no**, which means the only WSDLs available to Data Management Server are previously-generated WSDLs that are loaded at startup of Data Management Server. Requests to generate a WSDL or to upload a job with WSDL generation will return errors resulting in the job not uploading. Set the option to **single** to enable generation of a single WSDL per request, e.g. postJob request or genWSDL request for a single file. Set the option to **multiple** to enable generation of multiple WSDLs per request, e.g. genWSDL request for

Configuration Option	Description
	multiple job files or for entire directories of jobs. Note that generating a WSDL can be a time-consuming and resource-intensive process which depends on the parameters of the originating job. Also of note is that a malicious or uninformed user who sends a request to generate WSDLs for all jobs under the root directory can cause a severe degradation in server performance while the WSDLs are being generated; therefore, use the **multiple** option with caution.
DMSERVER/SOAP/WSDL/GEN_ON_GET	Specifies whether or not to allow WSDL generation as part of an HTTP getWSDL request. The default setting is **no**, which means that an error is returned when a WSDL is requested if: 1) it does not already exist, or 2) if it exists but no longer matches the job's **`mod.time`**. If set to **yes** in the situations described above Data Management Server attempts to generate the latest WSDL and if successful, returns the WSDL. This option is activated only when DMSERVER/SOAP/WSDL/GEN is set at **single** or **multiple**. If DMSERVER/SOAP/WSDL/GEN is set at **no**, this option is ignored and WSDLs will not be generated on HTTP getWSDL requests.
DMSERVER/SOAP/WSDL/RUN_IGNORE_MTIME	Specifies whether or not to ignore a difference between WSDL and job **`mod.time`** stamp when a runSVC request comes in. The default setting is **no**, which means that when a client sends a request to run a service based on a job, WSDL and Data Management Server determine that the WSDL no longer matches **`mod.time`** of the service job file, Data Management Server will return a SOAP Fault message '*Job Has Changed*'. The client knows then that the job has changed on Data Management Server and in response, will regenerate the WSDL, obtain it from Data Management Server, and rebuild

Configuration Option	Description
	the client against the latest WSDL. If the option is set to **yes**, Data Management Server will not check whether its WSDL **`mod.time`** matches that of the service job and will pass the received service request to the service process for execution. Note that this is the behavior of service requests based on generic WSDL.

Debug Services with SOAP Fault Elements and Log Files

When a client submits a SOAP request, and when the server encounters an error during the processing of that request, the SOAP response from the server will contain a <faultcode> element and a <faultstring> element. The faultcode tag categorizes the error. The faultcode provides a human-readable description of the error.

The Data Management Servergenerates the follows fault codes:

Error	Description
VersionMismatch	VersionMismatch pertains to the **SOAP Envelope element**. which is the root element for all SOAP messages. The VersionMismatch fault indicates that the recipient of a message did not recognize the namespace name of the Envelope element.
MustUnderstand	MustUnderstand indicates that the recipient of an element child of the Header element had a **`soap:mustUnderstand`** attribute but that element was not understood by the recipient.
SOAP-ENV:Client	Indicates the SOAP message did not contain all of the information that is required by the recipient. This could mean that something was missing from inside the Body element. Equally, an expected extension inside the Header element could have been missing. In either case, the sender should not resend the message without correcting the problem.
SOAP-ENV:Server	Indicates that the recipient of the message was unable to process the message because of some server-side problem. The message contents were not at fault; rather, some resource was unavailable or some processing logic failed for a reason other than an error in the message.

Other fault elements in the SOAP response, such as <*faultactor*>, <*detail*> etc.), do not indicate a specific reason or data element that triggered an error. These are more of a generic nature and are usually returned for any and all SOAP requests. Because Data

Management Server logs information related to processing errors, these optional SOAP elements are not used for error messaging. It is necessary to look at a log file to obtain details for problem determination. Depending on the nature of the problem, the error might be exposed in a server log or specific service process log. Refer to the Server Logs and Data Service Logs topics for additional information.

Also, the **nil** elements are currently unused for SOAP fault messaging. It is best not to refer to these elements for problem determination.

Manage the DFVSVC Process

- Configuration Options for DFWSVC
- Other Configuration Options (service.cfg)
- Unload Processes

Two processes are used to run services and jobs: **DFWSVC** and **DFWFPROC**. DFWSVC runs real-time data services, while DFWFPROC runs process services, batch and profile jobs. The following information explains each of these processes along with configuration options for each process.

A DFWSVC process runs real-time data services and is managed by the Data Management Server. The server tracks both idle and active processes and knows if any service jobs are loaded and waiting to run. When a new service job is received, the server first tries to finds an idle DFWSVC process. If one does not exist, the server looks for a DFWSVC process that does not have any jobs loaded. Finally, if the server does not find a process to reuse, a new process is started, if the configuration allows.

If a DFWSVC process is terminated, either because it crashed or was killed, the Data Management Server will notify you of the event and will log this event into server's log. If an idle dfwsvc process terminates, the server will log this event and start a new process when another data service request is received. You can set the maximum runtime for data services by using the DMSERVER/SOAP/DATA_SVC/MAX_RUNTIME configuration. Configuration options are explained in the following topic, Configuration Options for dfwsvc.

> **Note:** When processes are reused too often, performance may be reduced. You can specify the POOLING/MAXIMUM_USE option in the **app.cfg** file for Data Management Server that controls the maximum number of times a pooled process may be reused. After the pooled process has been used the specified number of times, it is terminated.

Configuration Options for DFWSVC

Following are configuration options that are available for the DFWSVC (dfwsvc.exe) process. Configure these options as necessary your environment. These configurations are set in **dmserver.cfg** located in *install-path* \etc.

Configuration Option	Description
DMSERVER/SOAP/DATA_SVC/IDLE_TIMEOUT	Specifies the number of seconds to allow a DFWSVC process to remain idle before it is terminated. The default setting is 0 indicating that there is no

Configuration Option	Description
	timeout. Negative values are ignored.
DMSERVER/SOAP/DATA_SVC/QUEUE	Specifies whether to queue real-time data service requests. If set to **yes** and all running dfwsvc processes are busy and Data Management Server is not allowed to start another one to handle a new service request, the request is put in a queue. As soon as a dfwsvc process becomes available, the request is handled. The default is **no** and in the above scenario, an error message is displayed.
DMSERVER/SOAP/DATA_SVC/MAX_NUM	Specifies the maximum number of real-time data services the SOAP server is allowed to run simultaneously. The default is ten. If a new service request would exceed the set limit and queue is not enabled, an error message is displayed. This option applies to the SOAP server, meaning the service requests are coming from a SOAP client. It does not apply to the WLP server or requests coming from a WLP client.
DMSERVER/WLP/DATA_SVC/MAX_NUM	Specifies the maximum number of real-time data services the WLP server is allowed to run simultaneously. The default is 10. If a new service request exceeds this limit, an error message is displayed. This option applies to the WLP server, meaning the service requests are coming from a WLP client. It does not apply to the SOAP server or requests coming from a SOAP client.
DMSERVER/SOAP/DATA_SVC/MAX_ERRS	Specifies the maximum number of service errors that can occur in a **dfwsvc** process before it is forced to terminate. The default is -1, meaning there is no limit.
DMSERVER/SOAP/DATA_SVC/MAX_REQUESTS	Specifies the maximum number of service requests a **dfwsvc** process is allowed to handle before it is forced to terminate. The default is -1, meaning

Configuration Option	Description
	this is no limit.
DMSERVER/SOAP/DATA_SVC/MAX_RUNTIME	Specifies the number of seconds to allow a data service to run a job and how long it has to produce a response (output data or an error). If a data service does not produce a response within the configured number of seconds, Data Management Server terminates the corresponding dfwsvc process and sends a SOAP Fault message to the client. The default is zero, meaning no timeout occurs. Negative values are ignored. Note that the timeout count may be off by approximately a second or two due to rounding up of counts less than a second (e.g. 1.5 = 2 secs).

Other Configuration Options (service.cfg)

The following table lists additional configuration options for **DFWSVC** and are set in **service.cfg**. When using these configurations, consider that similar configuration options in **app**.**cfg** are applied to a job first and then followed with configuration options from this file.

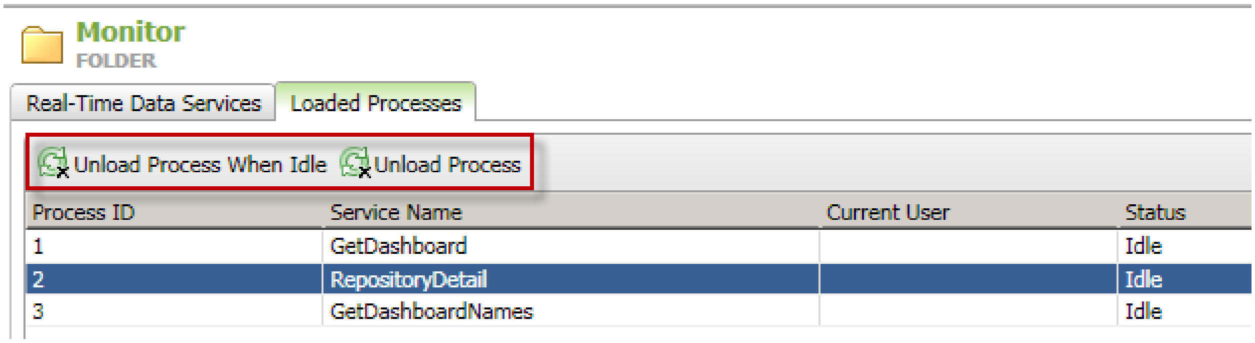
Configuration Option	Description
DATASVC/IGNORE_DECLARED_VARS	Options: Yes/No. This is used to ignore input / output variables that are declared in a data job. If set to yes (input and output variables are ignored) to allow for standard behavior prior to release 2.2, where all macros that a user passes into a data service make their way into the job, and the final values of all those macros are returned back to the user along with the output data from the service. This option is set at no by default, so only input variables that are declared on the job can be passed in. If any other input variable is passed in, that call will error out. Only the output variables declared on the job will be returned along with the output data from the service.
DATASVC/THREAD_STACK_SIZE	To configure options for the dfwsvc process on a non-Windows platform, use this option to set the stack size for each thread to use, in bytes. The default value is 1MB. This applies to non-Windows platforms only and is ignored in a

Configuration Option	Description
	Windows environment.
DATASVC/MAX_OUTPUT_BYTES	Use this option to control the amount of data (in bytes) the DFWSVC process is allowed to return to the SOAP client. The default value is 128MB (134217728 bytes). If the output data from DFWSVC exceeds the limit set in this configuration, the job is aborted and an error message "*Output data size limit exceeded*" is logged in the DFWSVC and Data Management Server logs and sent to the SOAP client. A value of 0 or less means output data size checking is disabled. **NOTE**: This option applies only when DFWSVC is invoked from a SOAP client request.

For a complete list of configuration options for Data Management Server, refer to the Configuration Options Reference for dmserver.cfg.

Unload Processes

To unload a service process, select the **Real-Time Data Services** folder in the navigation pane, and choose the **Loaded Processes** tab. Select one or more service jobs and choose **Unload process When Idle** or **Unload Process** to end the process immediately.



Process ID	Service Name	Current User	Status
1	GetDashboard		Idle
2	RepositoryDetail		Idle
3	GetDashboardNames		Idle

Real-Time Data Services - Unload Process Dialog

To unload processes for batch jobs, profile jobs and process services, see Unload Processes in the DFWFPROC topic.

Manage the DFWFPROC Process

The DFWFPROC process runs process services, batch jobs, and profile jobs. Process services are handled independently of batch and profile jobs, by a pooler. The Data Management Server requests a new process from the pooler and then notifies that process which service job to load and run with the specified input parameters, if any were sent with the job request.

> ***Note****: The default directories for process services, batch jobs and profile jobs are located in* install-path*var.*

Unlike the service process (DFWSVC), the server does not manage DFWFPROC processes. When a new process is requested, the server receives a new process if the pooler cannot find an idle process to use. A service job is then loaded for the process to run. If the process terminates for any reason, for example if the new service request exceeds the allowed limit, Data Management Server will send an error message about the problem. By default, the Data Management Server allows 10 service requests to run simultaneously. You can change the maximum number of requests by setting the number in DMSERVER/SOAP/PROC_SVC/MAX_NUM, found in the dmserver.cfg file.

For batch jobs, the Data Management Server starts a new DFWFPROC process and assigns a batch job to run. You will be notified whether or not the run started. Optionally, you can monitor the progress of the job. The server periodically checks if the job is still running, has finished (either successfully or with an error), or if the DFWFPROC process has terminated.

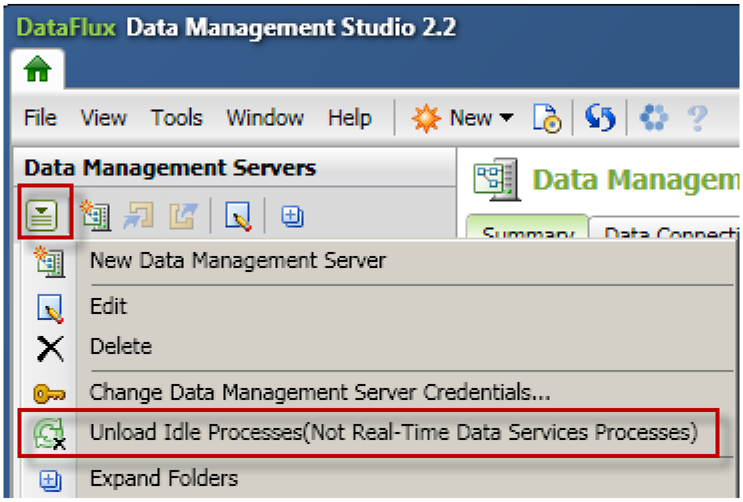
Following are configuration options available for DFWFPROC (dfwfproc.exe). For a complete list of configuration options for Data Management Server, refer to the Configuration Options Reference for dmserver.cfg.

Configuration Option	Description
DMSERVER/SOAP/PROC_SVC/MAX_NUM	Specifies the maximum number of real-time process services that Data Management Server is allowed to run simultaneously. The default is ten. If a new service request exceeds this limit, an error message is displayed.
DMSERVER/JOBS_MAX_NUM	Specifies the maximum number of batch and profile jobs that Data Management Server allows to run simultaneously (both batch and profile jobs are counted against the same pool). The default is 10. If a new job request is issued that exceeds the limit, an error message is displayed.
DMSERVER/SOAP/DATA_SVC/JOB_COUNT_MIN	Specifies the minimum number of instances of a given service job that must remain loaded. Once this number is reached, Data Management Server will not unload any more instances of that service job. Usage: = *count:job_file_name*
DMSERVER/SOAP/DATA_SVC/JOB_COUNT_MAX	Specifies the maximum number of instances of a given service job that can be loaded at the same time. Once this number is reached, Data Management Server will not load any more instances of that service job. Usage: = *count:job_file_name*
DMSERVER/JOBS_ROOT_PATH	Specifies the location of the root directory for the jobs and services

Configuration Option	Description
	sub-directories. The default object root directory is *install-path*\var. The sub-directories for jobs and services are: data services, process services, and batch jobs.

Unload Processes Spawned by DFWFPROC

You can unload idle processes spawned by DFWFPROC using the SOAP command UnloadProcesses, or by using the **Action Menu** in the Data Management Server toolbar:


DataFlux Data Management Studio 2.2
File View Tools Window Help New
Data Management Servers
Data Managem
Summary Data Connecti
New Data Management Server
Edit
Delete
Change Data Management Server Credentials...
Unload Idle Processes(Not Real-Time Data Services Processes)
Expand Folders

Data Management Server: Unload Idle (dfwfproc) Processes

This action unloads idle processes for process services, batch jobs and profile jobs. To unload processes for real-time data services, see Unload Processes.

Manage Repositories, Services and Jobs

Data Management Server jobs can be divided into two categories: real-time service types and batch job types. Real-time service types include data and process services, while job types include batch and profile jobs. Real-time services are performance-oriented and can take as little as a few seconds to complete, depending on the nature of the service job. On the other hand, batch jobs generally take longer to run, ranging in time from a number of seconds to several hours, depending on the nature of the job.

Refer to the Data Management Server User's Guide for information regarding running these services and jobs on the Data Management Server.

Real-Time Service Run Types

A real-time service is a request sent to the Data Management Server to run a job that includes specific data or variable inputs. After execution of the service request, the server should return the output data; however, in some instances the request may run indefinitely, causing the connection to time out. By default, logging is not available with Real-Time

Service runs, but server-level logging can be activated in certain situations. For more information, refer to the Data Service Logs topic.

There are two real-time service run types: data services and process services. Service jobs are stored in the **data_services** and **process_services** directories located in *install-path* \ var\.

Real-Time Data Services

A real-time data service, which was referred to as an architect service in previous versions of the server, uses the **dfwsvc** process to run. This process is outlined in the Server Processes topic. A data service can have an External Data Provider (EDP) node, in which you can pass input data into the service to be processed. If the data service does not have an EDP node, it might receive input data from another source, depending on the nature of the service. For example, a data service can be designed to retrieve data from a database. If you have existing architect services, you can run them as data services, without converting the old job files.

Real-Time Process Services

A real-time process service uses the **dfwfproc** process that runs a WorkFlow Engine (WFE), which can run process service jobs. Process services do not accept input data to be processed; they only accept input *parameters*. If you have existing architect services, you can run them as process services, without converting the old job files. For more information about the dfwfproc process, see Manage the DFWFPROC Process.

Batch Job Run Types

A batch job run occurs when you send a request to the Data Management Server to run a given batch or profile job. The job request might include inputs to process. Once the job finishes running, check the job log to obtain the final status of the run. All batch jobs are logged in dmserver.log. For more information, refer to the Batch Job Logs topic.

There are two batch job run types: batch jobs and profile jobs. These jobs are stored in *install-path*\ var\batch_jobs.

Batch Jobs

A batch job replaces the architect batch job that was available in previous versions of the server. Batch jobs use the **dfwfproc** process. You can pass inputs into batch jobs, but not any actual data for processing, unlike real-time data services. The inputs must be declared as part of the job or the job run will fail due to unknown inputs; an error message will be displayed. If you have existing architect bath jobs, you can run them as batch jobs, without converting the old job files.

Profile Jobs

Profile jobs are handled as repository objects; therefore, they must reside in the Data Management Repository. When you run a profile job, the server finds the job in the repository, starts a new **dfwfproc** process, and uses **ProfileExec.djf** to run the requested profile. **ProfileExec.djf** is installed with Data Management Server in the **repositories** directory where the Data Management Repository is located. If deleted or moved, profile jobs cannot run.

Unlike batch jobs, you cannot grant unique user permissions for profile jobs since they do not have associated object-level access control. See Manage Permissions.

Profile jobs from previous versions of the server **cannot** be run as-is; they must be converted and imported into the new server repository. For more information on upgrading profile jobs, see Upgrading Jobs, ACL Settings, and Command Permissions.

About Repositories

A repository is a directory that stores a database file or database connection information. You run jobs on the Data Management Server to access the data in the repository's database.

The Data Management Server enables access to one and only one repository. If a job attempts to access a repository other than the one that is currently enabled, the server sends an error messages to the server log file.

Each repository is defined by a repository configuration file. The **.rcf** file provides metadata definitions for Vault root directories and Unified Database connection parameters. Vault directories are not used by the Data Management Server. The Unified Database connection consists of a connection string and a table prefix. The connection string can identify a file in the repository, or a connection to a DBMS.

The **.rcf** file is located in the directory of the repository. The location of the directory is specified by default or by the configuration option BASE/REPOS_SYS_PATH, which is not set by default.

The default repository is *install-path***\etc\repositories**. The default repository configuration file in that directory is **server.rcf**. To enable a non-default repository, set the option BASE/REPOS_SYS_PATH in dmserver.cfg and restart the Data Management Server.

Related Topics

- Create a Repository

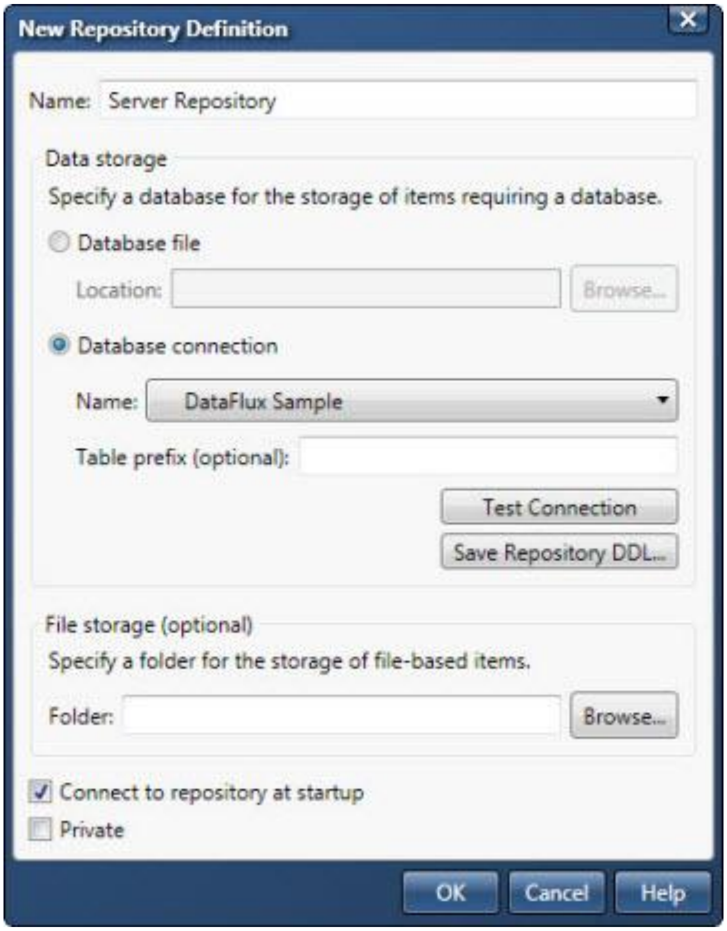
Create a Repository

Use these steps to create a repository in Studio and then copy the repository configuration file to the Data Management Server:

1. Click on the **Administration** riser bar in Studio.
2. In the **Repository Definitions** pane, click on the **New** button to create the new repository.
3. The New Repository Definition dialog appears. Enter the **name** of your repository in the Name field.
4. In the Data storage section of the dialog, you have two choices:

    a. Specify a **Database file** or a **Database connection** for your new repository. The Database connection must have been created earlier.

    b. To work with Data Management Studio objects that are stored as files, specify a **File storage** location using the **Browse** button. This physical path must be accessible to everyone who needs to access the data jobs, process jobs, queries, *.sas files (SAS code files), or Entity Resolution Output files (*.sri files) that will be stored there.



New Repository Definition

Name: Server Repository

Data storage

Specify a database for the storage of items requiring a database.

○ Database file

Location:

● Database connection

Name: DataFlux Sample

Table prefix (optional):

File storage (optional)

Specify a folder for the storage of file-based items.

Folder:

☑ Connect to repository at startup

□ Private

Data Management Studio Repository Definition

5. Select or deselect the **Connect to repository at startup** checkbox, and uncheck the **Private** checkbox. (If you select Private, the repository is stored to a location on the local machine and cannot be viewed by others.)

6. Click **OK** to create the new repository.

   Note: The location of the repository file in Studio is: *drive*:\Program Files (x86)\DataFlux\DMStudio\version\etc\repositories.

7. After the repository is created, copy the file from Studio to the Data Management Server using one of the following methods.

Database file:

If the Data Storage area for the repository specifies a DBMS location, perform the following steps.

1. Copy the repository configuration file (.RCF file) from Data Management Studio. Data Management Studio stores .RCF files according to their status:

   Private Repository:

***install-path*\repositories\*repository-name*.rcf**

Public Repository:

***install-path*\etc\repositories**

2. Copy the repository configuration file into the desired directory on the host of the Data Management Server. The default repository is:
   ***install-path*\var\repositories**

3. Create an ODBC data connection with the same name that is used in the .rcf file.

File-Based Repository:

If the Data Storage area for the repository specifies a file location, such as an SQLite file, perform the following steps.

1. Copy the repository data file from Data Management Studio and upload it to a convenient location on the Data Management Server.
2. Copy the repository configuration file (.RCF file) from Data Management Studio.
3. Upload the repository configuration file to the Data Management Server.
4. Update the repository configuration file to point to the repository data file from Step 1.

Related Topics

- About Repositories

Configure Jobs and Services

To modify configuration that controls data processing (data that is processed by data and process services, and by batch and profile jobs) modify the **app.cfg** file. For more information on the **app.cfg** file, see the Configuration topic of the *DataFlux Data Management Studio Installation and Configuration Guide*.

You can use configuration files to modify how data services and batch jobs run. Located in ***install-path*** \etc, the files are:

1. **service.cfg**- used to run real-time data services, and
2. **batch.cfg** - used to run batch jobs.

Note: There is an order of precedence for configuration settings. In general, first a setting is determined by the Advanced Properties of a node in the job or real-time service. In the absence of a setting, the value is set by the corresponding entry in the **macros.cfg** file. If there is no specific setting, Data Management Server then obtains the setting from the appropriate configuration file. If the value has not been set, Data Management Server will use the default value.

Grant Job Permissions

You can configure user permissions for batch jobs using the Data Management Server interface in Studio. This procedure is outlined in *Granting Permissions from a Job List* in Manage Permissions. Profile jobs do not have security functions like batch jobs, so they cannot be secured at the job level. You can still grant user permissions for profile jobs at the user or group level.

Configure Bulk Loading in Monitor Jobs

Bulk loading enhances the performance of jobs that monitor business rules, when those jobs include row-logging events. You can optimize performance for your implementation by changing the number of rows in each bulk load . By default, the number of rows per load is 1000. You can change the default value in the app.cfg option MONITOR/BULK_ROW_SIZE.

The business rules monitor creates and executes temporary jobs, and those jobs are normally kept in memory. When a directory is specified for temporary jobs, the Monitor creates temporary jobs in that location and leaves them in place after the job is complete. To specify a directory for temporary jobs, create the directory and set the path of that directory as the value of the app.cfg option MONITOR/DUMP_JOB_DIR. By default, this option is not set and the Monitor does not store temporary jobs on disk.

Define Macros

The **macros.cfg** configuration file defines macro values for substitution into batch jobs, and overrides predefined values. This file is located in the install-path/etc. Each line represents a macro value in the form KEY = VALUE, where the KEY is the macro name and VALUE is its value. For example, on a Windows system:

```
INPUT_FILE_PATH = drive:\files\inputfile.txt
```

On a UNIX system:

```
INPUT_FILE_PATH = /home/dfuser/files/inputfile.txt
```

The examples above set the macro value INPUT_FILE_PATH to the specified path. This macro is useful when you are porting jobs from one machine to another, because the paths to an input file in different platforms may not be the same. By using a macro to define the input file name you do not need to change the path to the file in the batch job after you port the job to UNIX. Add the macro in both the Windows and UNIX versions of the **macros.cfg** file, and set the path appropriately in each.

> **Note:** The **etc/macros** subdirectory may contain files with a **.cfg** extension. If one or more files exist, they will be read in alphabetical order **before** the **macros.cfg** file is read.

If Studio users are using system and user-created macros, you must create a combined macro file to be able to use the macros in Data Management Server. For more information on macros, see the *DataFlux Data Management Studio Online Help*.

Declare Input and Output Variables for Data Services

When running data services in previous releases of Data Management Server, the macros that were passed into a data service were the only ones that would be returned from the service with the final values set. Also, any macro variable was allowed to be passed to the service, whether or not it was actually being used by the service.

Beginning with release 2.2, input and output variables for data services behave similarly to variables of process services and batch jobs. Specifically, only input variables that are declared in a data service job can be passed in, and only final values for declared output variables will be returned. If a variable that was not declared as input is passed into a data service, an error will be returned. To revert to behavior prior to release 2.2, set the following configuration option in the **`service.cfg`** file:

```
DATASVC/IGNORE_DECLARED_VARS = yes
```

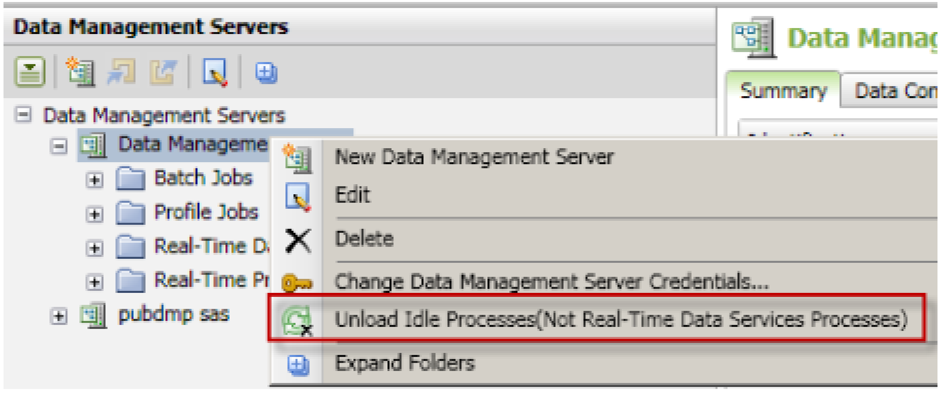
Update Macros

Each service and job process on Data Management Server reads configured macros as it is started. When a macro changes, you can update the macro on Data Management Server without having to restart the server. To update macros on Data Management Server, use one of the following procedures.

Process Services, Batch and Profile Jobs

For process services, batch jobs, and profile jobs using the DFWFPROC process:

1. In Data Management Studio, select a Data Management Server name from the list of servers in the left pane.
2. Right-click on the Data Management Server and select **unload idle processes** from the drop-down menu.



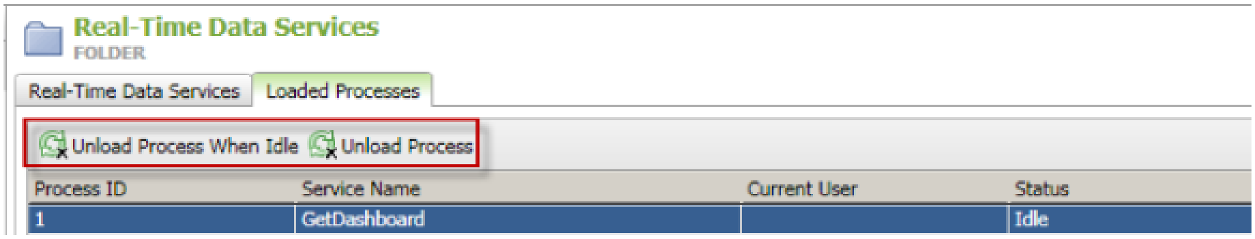
Real-Time Data Services

For real-time data services using the DFWSVC process:

1. Select a Data Management Server from the left pane in Data Management Studio.
2. Select the **Real-Time Data Services** folder under the Data Management Server instance.

3. In the right pane click on the **Loaded Processes** tab.

4. Select all of the processes under Process ID, and click one of two buttons depending on the status of the job: **Unload Process When Idle**, or **Unload Process**:



Process ID	Service Name	Current User	Status
1	GetDashboard		Idle

Run Jobs From a Command Line

Use the **dmpexec** command to execute profiles, data jobs, or process jobs from a command line. The dmpexec command is located in *install-path***bin**. The following table defines the options for the dmpexec command.

Option	Purpose
-c *file*	Read a configuration file to set option values that are specific to the job or command, including authentication options (see the -a option.)
-j *file*	Execute the job in a file.
-l *file*	Write job log messages to a file.
-i *key=value*	Set the input variable (*key*) to a value before running the job.
-o *key=value*	Set a server option to a value.
-b *key=value*	Set a job option to a value.
-a	Authenticate using an Authentication Server. This option is required for domain-enabled connections. To successfully authenticate, you need to specify options that identify the location of the Authentication Server, specify a user name for that server, and specify a password. See Configure Authentication for dmpexec.

Note: You can use the -i, -b, and -o options multiple times to set multiple values.

Configure Authentication for dmpexec

When you specify the **-a** option in the dmpexec command, the Data Management Server requires three server configuration options. The configuration options specify the location of the Authentication Server, a user name that is registered on the specified Authentication Server, and the password that validates the user name:

BASE/AUTH_SERVER_LOC=*network-path*:*port*
Specifies the Authentication Server that verifies the credentials that are submitted with the command.

BASE/AUTH_SERVER_USER=*user-name*Specifies a user name that is registered on the specified Authentication Server.

BASE/AUTH_SERVER_PASS=*password*
Specifies the password that is associated with the user name.

You can set default values for these options in the configuration file dmserver.cfg. You can also specify these options in a configuration file that you create specifically for a given job or command, using the **-c** option.

Return Codes

The following table reflects status codes for jobs run from the command line:

Return Code	Reason
0	Job is still running
1	Job has finished successfully
2	Job has finished with errors: general error (see job log file)
3	Job has finished with errors: process aborted
4	Job has finished with errors: job was canceled

Technical Support

- Best Practices
- Frequently Asked Questions
- Troubleshooting

Best Practices

Use a System Data Source Rather Than a User Data Source

Add the new data source as a System data source name (DSN), rather than a User DSN, so it will be available to all users of the machine, including Microsoft® Windows NT® services. This only applies to Windows systems.

Use the Data Connections Riser to Configure Data Sources on Windows Systems

When developing DataFlux Data Management Studio (Studio) jobs or services, use the Data Connections riser to set up and store login credentials for any Open Database Connectivity (ODBC) data source. The Studio client must be on the same machine as the server.

Use global variables within jobs and services to accept or retrieve data. Using global variables increases the flexibility and portability of Studio jobs and services between data sources.

The saved credentials do not have to be entered each time the job is run, and that information can be used by any DataFlux application.

For UNIX systems, the connection information is saved to a file in the `/$HOME/.dfpower/dsn` directory.

Plan Your Security Model Based on Business Needs

The Data Management Server application is a network resource that is used to access and modify your data. A well-planned security model is based on usage policy, risk assessment, and response. Determining user and group usage policies prior to implementation helps you to minimize risk, maximize utilization of the technology, and expedite deployment.

For more information, see Implement a Security Policy.

Creating ODBC Connections Using Saved Credentials

When using ODBC connections, it is suggested that you create ODBC connections using saved credentials. Otherwise, if credentials are required, your jobs will fail. Another option is to use the Dataflux Authentication Server for managing credentials on your system.

Managing Jobs

The user interface for the Data Management Server is integrated into the Studio client and can be accessed via the Data Management Server riser. The Data Management Server riser is the best way to manage jobs.

Frequently Asked Questions

How do I change the default temp directory?

Update the path for the base/temp configuration option, in the **app.cfg** file located in the ***install-path*\etc** directory.

How do I connect to a database?

Data Management Server connects to databases through ODBC or through the optional Federation Server. To add a data source using ODBC, use the ODBC Data Source Administrator provided with Windows, or use the **dfdbconf** command in UNIX. In Windows, start the Data Management Studio client and navigate to the **Information Riser Bar**. In the **Overview** pane, click **Documentation** to list the DataDirect Connect ODBC Help. For more information, see Configure ODBC Connections.

For more information on the Federation Server, see the *Federation Server Administrator's Guide*.

What is a Data Management Server?

A Data Management Server is a service-oriented architecture (SOA) application server that allows you to execute batch or profile jobs created using the Studio design environment on a server-based platform. This could be Microsoft Windows, Linux, or nearly any other UNIX option.

By processing these jobs in Windows or UNIX, where the data resides, you can avoid network bottlenecks and can take advantage of performance features available with higher-performance computers.

In addition, existing batch jobs may be converted to real-time services that can be invoked by any application that is Web service enabled (for example: SAP®, Siebel®, Tibco®, Oracle®, and more). This provides users with the ability to reuse the business logic developed when building batch jobs for data migration or loading a data warehouse, and apply it at the point of data entry to ensure consistent, accurate, and reliable data across the enterprise.

What is the difference between Data Management Server – Standard and Data Management Server – Enterprise?

The Data Management Server – Standard supports the ability to run batch Studio jobs in a client/server environment, as well as the ability to call discrete DataFlux data quality algorithms from numerous native programmatic interfaces (including C, COM, Java™, Perl, and more). The Data Management Server – Standard allows any Studio client to offload batch Studio profile and batch jobs into more powerful server environments. This capability frees up the user's local desktop, while enabling higher performance processing on larger, more scalable servers.

The Data Management Server – Enterprise edition has added capability allowing you to call business services designed in the Studio client environment or to invoke batch jobs using SOA.

How do I move a batch or profile job to UNIX so it can be processed by a Data Management Server?

With Data Management Server installed as part of Studio, use the Data Management Server Riser Bar to connect to the desired server and select the job or service to be uploaded. You can also use this method to test real-time services on your Data Management Server.

What SOAP commands are recognized by Data Management Server?

For a complete list of SOAP commands recognized by Data Management Server, see SOAP Commands and WSDL Types.

How do I add an additional driver for the data sources?

Data Management Server is compatible with most ODBC compliant data sources. With the optional addition of the Federation Server, additional data source driver options are available, including drivers that are written for native access to a number of popular database sources. When using ODBC drivers, DataFlux recommends using the DataFlux provided ODBC drivers instead of client-specific drivers provided by the manufacturer. Limited support will be available for implementation and problem resolution when a client implements a driver not supplied by DataFlux.

For a complete list of supported ODBC and other drivers, see the *Federation Server Administrator's Guide*.

I can't see my saved job even though it's saved in a directory I have access to. Where is it?

In Windows, a job that will be run through Data Management Server must be saved in a location that does not use mapped drives. A Windows service is not able to access mapped drives, even if the service is started under a user account that has those drives mapped.

Are there any characters that are allowed in job names?

Data Management Server job names can include alphanumeric characters, along with the following characters:

. , [ ]{ }( ) + = _ - ^ % $ @ ! '

Is there a limit to the size of job and service names?

Data Management Server limits all job and service names to less than 8,192 bytes. If you create a job or service name that is greater than this limit, Data Management Server will reject your request and return the following SOAP Fault message to the SOAP client: *Invalid SOAP Request Contents*.

Can I run a UNIX shell command from a Data Management Server job?

Yes, the execute() function allows you to run a program or file command from the shell. For example, the following code allows you to modify the default authorizations of a text file.

To run the command directly, type:

```
execute("/bin/chmod", "777", "file.txt")
```

To run the command from the UNIX shell, type:

```
execute("/bin/sh", "-c", "chmod 777 file.txt")
```

Why is my job failing to read SAS Data Sets on AIX?

In order to access SAS® data sets on AIX®, you must have AIX 5.3 with patch level 6 installed on your system.

How are SAS Data Types Converted When DataFlux Software Reads or Writes SAS Data?

Automatic data-type conversions will take place when a data job reads or writes SAS data sets. For more information, see the *DataFlux Data Management Studio Online Help*.

How can I configure the directory where data source name (DSN) and saved connection files are stored?

The Data Access Component (DAC) options are configured in the **app.cfg** file. The following table lists the appropriate key that should be added to the `app.cfg` file and the default location where the file will be saved if the key and value pair is not specified:

Setting	Default Value
DAC/DSN	*install-path* \etc\dftkdsn\
DAC/SAVEDCONNSYSTEM	*install-path* \etc\dsn\

For more information on the **app.cfg** file, see the *Configuration* section of the *DataFlux Data Management Studio Installation and Configuration Guide*.

Are there any special considerations for ODBC drivers using the wire protocol?

DataDirect provides a number of wire protocol ODBC drivers that communicate directly with a database server, without having to communicate through a client library. If these drivers are available at your site, then they are available from the **Drivers** tab of the ODBC Data Source Administrator dialog. For more information on using wire protocol drivers and the special considerations that apply, see the *Frequently Asked Questions* section of the *DataFlux Data Management Studio Online Help*.

Can I use an older version of the Blue Fusion library, so I can continue using previously generated matchcodes?

Yes, you can use the QKB/COMPATVER option to specify which version of Blue Fusion will process the Quality Knowledge Base definitions. The QKB/COMPATVER option is defined in the **app.cfg** file and can be set to *unity21* if you want to use functionality new to Studio or *dfpower82* if you want to use functionality that existed in dfPower version 8.2.

For more information on the QKB/ALLOW_INCOMPAT option and the app.cfg file, see *Configuration Files* in the *DataFlux Installation and Configuration Guide*.

Where is data source name information stored?

On Windows systems, data source name (DSN) connections are located in the ***install-path*\etc\dftkdsn** directory.

On UNIX systems, DSN connections are defined in **odbc.ini** located in the **etc** directory.

How are real-time service log files created and handled?

For both real-time services (data or process), once a process to run a service is started (either DFWSVC or DFWFPROC), the process will retain its log file, even if it runs other service jobs. The Data Management Server log file will contain the name of the corresponding process log file for each service request handled. See the Server Logs topic for detailed information regarding activation and use of log files that are available with Data Management Server.

Troubleshooting

- Data Management Server
- Job-Related Error Messages
- QKB
- ODBC
- Security
- ActiveX

Data Management Server

Unable to start Data Management Server:

If you are not able to start Data Management Server and the log file lists a dfwlpListenAttr_connattr(wlp) failure, ensure the default ports are not being used by another application. Data Management Server uses two ports. By default, SOAP connections are handled on port 21036 and WLP connections are handled on port 21037. If either of the default ports are being used by another process, assign that Data Management Server process to an unused port.

Data Management Server does not start on my Windows or UNIX system, what should I do:

Check the Data Management Server log file for errors:

If the Data Management Server startup sequence proceeds far enough, a server_logs directory is created in the **var** directory. The name of the directory will include a time-stamp of when the Data Management Server instance was started accompanied by the process id (pid) of the instance. In this directory, you will find a Data Management Server log file, named **dmserver**. Check the log file for error messages. Refer to the Server Logs topic for further information regarding the log files, logging thresholds, and the various types of messages associated with the logs.

Check for DataFluxDMS and SAS Application errors:

- On Windows systems:
  1. Open the Windows Event Viewer.
  2. Select the Application event type.

3. Click the Source column, to sort the events based on the type of source.
4. Search the Source column for "DataFluxDMS". There will typically be two error events logged for each time period. One of the errors will not contain any useful information. The other error will contain details about why Data Management Server did not start.
5. Next, search the Source column for **SAS**. If the message is similar to the following, go to the indicated log file to see the details of the error:

   ```
   WARNING: Messages have been logged to the file named
   'C:\Documents and Settings\LocalService\Application
   Data\SAS\LOGS\DFINTG~1.EXE.1854.21CBDA9C.log'
   ```

- On UNIX systems:

  The error messages will be written to **stdout** of the shell from which the Data Management Server was started.

Data Service Error, Process Service Error, or Batch Job Error: Server Processes DFWSVC/DFWFPROC fail to start -or- Out of Memory error in Windows when launching server processes.

Server processes fail to start and Windows displays an error message:

```
The application failed to initialize properly (0xc0000142). Click
on OK to terminate the application.
```

In addition, it is possible that one of the following messages appears in the Data Management Server logs:

```
Data Service error: failed to start service process: 1 - Child
failed to contact server process. Failed to start base services,
rc=1 (Error loading dependency library).

Process Service error: Failed to get process, errorCode=2
(Process 'HOST:ADDR' exited unexpectedly.)

Batch Job error: failed to get process; err: 0 - Process
'HOST:ADDR' exited unexpectedly.
```

Windows is unable to start a new process, either DFWSVC or DFWFPROC, and an error message is logged in the Data Management Server logs. This issue can happen on a Windows desktop as well as a server. Check the Windows event log for application and system error messages regarding errors for the DFWSVC and DFWFPROC processes. If the Windows event log does not reflect entries for DFWSVC and DFWFPROC but these errors appear in the Data Management Server logs, it is likely that the failure to start processes is caused by the Windows operating system already running too many internal processes for the Data Management Server to start new server processes.

The behavior happens when the Windows system runs out of desktop heap, specifically, the desktop heap in the WIN32 subsystem becomes depleted. To free system resources, stop as many non-essential applications and processes as permissible and try to run the jobs again on the Data Management Server. If the errors persist, you may need to make a minor change in the Windows registry to increase the **SharedSection** parameter of the **SubSystems** key in **HKEY_LOCAL_MACHINE** as recommended in the following Microsoft Support article:

KB 126962: "Out of Memory" error message appears when you have a large number of programs running

Additional information can be found in these articles:

KB 184802: User32.dll or Kernel32.dll fails to initialize

KB 824422: Unexpected behavior occurs when you run many processes on a computer running SQL Server

"Required openSSL dlls were not found" message when starting Data Management Server

The required dlls from the openSSL installation were placed into a directory other than /bin during installation. Copy the dlls to the /bin directory and restart the server.

Job-Related Error Messages

The repository is newer than this client

While running a profile job, if you get a message similar to, *The version of repository <ReposName> is newer than this client*, then someone at your site has a newer version of Data Management Studio than you do and has upgraded the repository. Contact your site administrator about upgrading your Data Management Studio software.

SQL lookup job fails on an UNIX system using the Driver for BASE:

The Driver for BASE does not allow data sets to be created that cannot be read by MVS SAS. Therefore, if you have Driver for SAS files that contain mixed case or uppercase letters that cannot be accessed on UNIX systems, you will need to rename the file to all lowercase letters. Files created in previous versions of the product that contain mixed case or uppercase letters may also need to be renamed using lowercase letters. Once the files are renamed, they can then be accessed in jobs using any case. For example, the file may be named lookupsource. In jobs, you can reference LOOKUPSOURCE, lookupsource, or LookUPSoUrCe, just to name a few.

dfIntelliServer with Data Management Server on a UNIX Platform: 'DFC_[*NODENAME*]': Data flow - Step failed to initialize..'

You receive an error when running a job on Data Management Server using dfIntelliServer distributed nodes: *'DFC_[NODENAME]': Data flow - Step failed to initialize*. This error message is returned because the home environment variable for dfIntelliServer is not set on the Data Management Server. The $DFCLIENT_HOME environment variable must be configured so Data Management Server can load the dfIntelliServer client. To set the $DFCLIENT_HOME variable:

1. Navigate to the /**client**/**bin** directory of the dfClient installation path.
2. Run the following command: **./dfenv** which will return: *$DFCLIENT_HOME is not set*.
3. Run one of the following commands (based on your operating system):
   - eval `./dfenv -x32 sh`
   - eval `./dfenv -x64 sh`

  - eval `./dfenv sh`

4. Rerun **./dfenv** to verify that the $DFCLIENT_HOME is now set.

   ⚠ When setting the path to dfclient.cfg in **app**.**cfg**, *do not* surround the value for DFCLIENT/CFG in quotes. This can also cause operation errors between Data Management Server and dfIntelliServer.

When I try opening a job log from Data Management Server Manager, I get the following error:

> *Error occurred while attempting to retrieve job log: SOAP-ENV:Client:UNKNOWN error [or Timeout]*

This occurs on some configurations of Microsoft Windows Server® 2003 when the log file is greater than 32KB. A workaround for this problem is to set the following configuration value in the `dmserver.cfg` file. This should only be necessary for Data Management Server running on Windows Server 2003, and only if you experience this problem.

```
DMSERVER/LOG_CHUNK_SIZE = 32KB
```

Unable to run batch job and receiving the following error:

> *Batch Job error: failed to get process; err: 0 - Process 'HOST:ADDR' exited unexpectedly.*

This issue is addressed in the Data Management Server section above. See Data Service Error, Process Service Error, or Batch Job Error: Server Processes DFWSVC/DFWFPROC fail to start.

An error occurs when running an address verification (world) job on Linux server:

```
Error message:()
2011-11-10T10:52:11,303 INFO [00001789] - Node DATAFLOW_0 started.
2011-11-10T10:52:11,390 ERROR [00001793] - Unknown locale name
```

The job was set up using Address Doctor v4 which is no longer supported. Convert the job to use the new Address Verification (World 2) node. Run the job using Address Doctor v5 and the error does not appear. Data Management Server supports Address Doctor v5

QKB

Blue Fusion cannot process new Quality Knowledge Base definitions:

This occurs when a Quality Knowledge Base (QKB) is loaded which uses definitions that are newer than what the current Blue Fusion engine provides. By default, Blue Fusion will attempt to load the definitions, but will issue warnings before loading them. If the definitions include instructions that Blue Fusion cannot process, the instructions will be ignored and an error will be displayed. This could result in unwanted results.

The QKB/ALLOW_INCOMPAT option can be used to specify whether or not to allow incompatible QBK definitions to be processed by Blue Fusion. The option is defined in the `app.cfg` file and allows you to choose to either stop processing or allow the incompatibility and continue processing the definitions.

For more information on the QKB/ALLOW_INCOMPAT option and the app.cfg file, see *Configuration Files* in the *DataFlux Data Management Studio Installation and Configuration Guide*.

Job with Custom Scheme Fails to Run

A job with a custom scheme that fails to run will produce an error similar to the following:

```
0817_11:17:40.691 ERROR Node DATAFLOW_0 error: 3: BlueFusion
Plugin - Blue Fusion load scheme 'frfra001.sch.bfd' failed:
BlueFusion Plugin - Blue Fusion error -400: BlueFusion - Cannot
open file "frfra001.sch"..
0817_11:17:40.694 INFO Job terminated due to error in one or more
nodes.
```

You must ensure that:

1. the Quality Knowledge Base (QKB) you are using on the Data Management Server is an exact copy of the QKB used on Studio, and
2. the name of the scheme is typed correctly, as it is case sensitive.

To copy the QKB from Microsoft Windows to UNIX, use FTP or Samba mappings. You must restart the Data Management Server service and retry the job. On some UNIX systems, there is a case sensitivity issue with the schemes.

Once you copy the QKB over to the UNIX server, make sure that the name of the scheme is modified to all lowercase letters. It is located in the **qkb** directory, under **scheme**.

Licensing

Locale Not Licensed

If your job has a locale selected that you do not have listed in your license, you will get an error message similar to the following:

```
Error message DT engine: 2::ERROR::-105:Local     English [US]
not licensed
```

You must contact DataFlux Customer Support to update your license with the new locale. Also verify that the data file for that locale is located in the `/locale` folder of the QKB install location.

Node Not Licensed

An error message similar to the following can occur when the user has more than one copy of the license file or a single license file that does not support the node:

```
Failed to create step: Couldn't instantiate step 'SOURCE_ODBC'.
It is not an available step. The node may not be licensed, or the
plugin for the node may be unavailable.
```

The license file must exists in the *drive*`:\Program Files\DataFlux\DMServer\`*(Undefined variable: MyVariables.instance-name)*`\license` directory and be specified in the app.cfg file.

ODBC

SQL Server ODBC Driver on Windows platform

If you have an ODBC DSN that uses the *Windows* SQL Server driver, replace that DSN with the *DataFlux* 32 or 64-bit SQL Server Wire Protocol driver. Using the Windows SQL Server driver can cause problems when creating a repository.

> **Note**: Use a 32 or 64-bit driver depending on the installation platform of Data Management Server. Access the drivers from the Control Panel > ODBC Data Sources. The DataFlux drivers are listed in the **Drivers** tab.

Teradata and Informix ODBC drivers fail to load on Solaris x86 platform

DataDirect doesn't currently provide Teradata or Informix drivers for Solaris x86.

Teradata ODBC driver fails to load on Linux

The directory containing the Teradata client libraries needs to be in your LD_LIBRARY PATH. The exact path will vary depending on your Teradata client version.

Security

401 Unauthorized

If a user is not authenticated, there will be an HTTP error, 401 Unauthorized. This could mean that the user entered an invalid user name and password credentials, or the user account has not been created.

403 Forbidden

When a user receives the HTTP error, 403 Forbidden, it means that they do not have authorizations to execute a particular Data Management Server command. See Manage Permissions for additional information.

ActiveX

ActiveX Control Required to View Help Files

In Microsoft Internet Explorer® 6.0 and later, ActiveX controls are sometimes blocked for download. Security for ActiveX content from CDs and local files can be changed under Internet Options. Use the following procedure to change security for ActiveX, also known as active content:

1. In Internet Explorer, click **Tools** > **Internet Options**.
2. On the **Advanced** tab, under **Security**, select **Allow active content from CDs to run on My Computer**, and **Allow active content to run in files on My Computer**.

Appendixes

Configuration Options Reference for dmserver.cfg

The table below lists the possible configuration options for Data Management Server. To modify configuration that controls server operation, edit the **dmserver.cfg** file in ***install-path*\etc**.

After making changes to any configuration files, you must restart the server.

Configuration Options

The following table lists the available configuration options for Data Management Server.

Configuration Option	Description
BASE/AUTH_SERVER_LOC	Specifies the fully-qualified network path and port number of the Authentication Server. The syntax of the value is *path*:*port*. This option is valid only when you also specify the option DMSERVER/SECURE=YES. See Enable an Authenticaqtion Server. This value can also be used to run jobs from a command line, using dmpexec. This value is overridden when a dmpexec command provides a job-specific value.
BASE/AUTH_SERVER_PASS	Optionally specifies a default password that is submitted to the Authentication Server when you use the -a option in a dmpexec command. When you set this option in dmserver.cfg, the value is overriden when thedmpexec command provides a job-specific value.
BASE/AUTH_SERVER_USER	Optionally specifies a default user name that is submitted to the Authentication Server when you use the -a option in a dmpexec command. When you set this option in dmserver.cfg, the value is overriden when thedmpexec command provides a job-specific value.

Configuration Option	Description
DMSERVER/CHILD/LISTEN_HOST	Specifies the hostname or IP address to which Data Management Server must bind for **dfwsvc** child process connections. By default, this option is set to localhost. For more information on binding to a hostname or IP address, see DMSERVER/SOAP/LISTEN_HOST.
DMSERVER/CHILD/LISTEN_PORT	Specifies the port on which Data Management Server listens for connections from dfwsvc child processes. This option defaults to a dynamic available port. If this option is specified and you are running multiple instances of the server on the same machine, this port must be unique for the ports, both specified and default. For more information on the default ports, see DMSERVER/SOAP/LISTEN_PORT and DMSERVER/WLP/LISTEN_PORT.
DMSERVER/IPACC/ALL_REQUESTS	Controls access to all SOAP requests based on the client's IP address. By default, this option is disabled. For more information, see Control Access by IP Address.
DMSERVER/IPACC/NOSECURITY	Allows or denies to specified IP addresses the ability to bypass all security checks on the Data Management Server.
DMSERVER/IPACC/POST_DELETE	Controls access to posting and deleting SOAP requests based on the client's IP address. This includes uploading new objects, such as jobs and services, to Data Management Server and deleting existing objects from the server. By default, this option is disabled.
DMSERVER/JOBS_HISTORY_MAXAGE	Defines a retention period, in seconds, for the history items that are generated by batch and profile job run instances. After the completion of a job, and after the expiration of the specified time period, the Data Management Server purges the job's history from memory. The server also deletes any corresponding log files, statistics files. If the JOBS_KEEP_HISTORY option is enabled, a history record is also deleted from history database. The default value of the MAXAGE option is -1, which specifies that history items are

Configuration Option	Description
	never purged. Note that jobs can delete their own log and statistics files by submitting the SOAP command DeleteJobLog.
DMSERVER/JOBS_KEEP_HISTORY	A value of YES specifies that the histories of job run instances are retained across server restarts. The default value is NO.
DMSERVER/JOBS_MAX_NUM	Specifies the maximum number of batch and profile jobs that Data Management Server allows to run simultaneously (both batch and profile jobs are counted against the same pool). The default is 10. If a new job request is issued that exceeds the limit, an error message is displayed.
DMSERVER/JOBS_NO_STATE	Specifies whether Data Management Server allows batch and profile jobs to generate state files for their runs. The default value is **no** which means that jobs are allowed to generate state files (if configured to be generated). If set to **yes**, the server will not allow jobs to generate state files. If you submit a request to run a job and generate a state file, the server will return a SOAP Fault message *State Generation Not Allowed*.
DMSERVER/JOBS_ROOT_PATH	Specifies the location of the root directory for the jobs and services sub-directories. The default object root directory is *install-path*\var. The sub-directories for jobs and services are: data services, process services, and batch jobs.
DMSERVER/LOG_CHUNK_SIZE	Controls the size of each log file or statistics file chunk that is sent back to the client from the **getJobLog** request. For log file, this option controls the number of characters per chunk. For statistics files, this option controls the number of bytes per chunk. The default value is 512K.
DMSERVER/NO_WORK_SUBDIRS	Specifies whether or not to create log subdirectories for each Data Management Server instance. The default value is no which means that all log files are created in subdirectories under the default directory,

Configuration Option	Description
	server_logs, or an alternate directory specified in the DMSERVER/WORK_ROOT_PATH option. This option should be set to **yes** only in special cases, as it creates numerous log files (regular logs as well as debug logs for all server processes) for all run instances in a single directory. This makes it difficult to determine which jobs and services log files belong to which server run instance and corresponding log file. Each run instance of each process (server, dfwfproc, and dfwsvc) gets its own, unique log file. Therefore, each new Data Management Server run instance has to have its own log file, while pre-existing log files, if any, are renamed.
DMSERVER/SECURE	Specifies whether the Data Management Server security subsystem is needed. The default is **no** which means that subsequent *secure* configuration directives are ignored. If set to **yes**, other configuration options are required to properly secure the server, including setting the Authentication Server connection string via the BASE/AUTH_SERVER_LOC option.
DMSERVER/SECURE/DEFAULT_ACE_PUBLIC	Specifies what ACE is set for the PUBLIC group when Data Management Server creates a default ACL for an object. A default ACL is created when an object is uploaded to Data Management Server or when an existing object on the Data Management Server is accessed and a corresponding ACL does not exist. By default, an ACE is not created for the PUBLIC group, which implies **inherit** access for that group. The possible values for this option are: **allow** and **deny**. Any other value is treated as **inherit**.
DMSERVER/SECURE/DEFAULT_ACE_USERS	Specifies what Access Control Entry (ACE) is set for the USERS group when Data Management Server creates a default access control list (ACL) for an object. A default ACL is created when an object is uploaded to Data Management Server or when an existing object on the server is accessed and has no corresponding ACL. By default, no ACE is created for the USERS group, which implies **inherit** access for that group. The possible values for this option are: **allow**

Configuration Option	Description
	and **deny**". Any other value is treated as **inherit**.
DMSERVER/SECURE/GRP_ADMIN	Specifies the name of the Data Management Server administrator group. If this option is defined, the group must exist on the Authentication Server. If this option is not defined or the group does not exist on the Authentication Server, Data Management Server returns an error during startup if running in secured mode.
DMSERVER/SECURE/GRP_ALLOW	Specifies the name of the group whose members are allowed access to Data Management Server. This is an optional setting to provide a convenient way to exclude most users and allow only a few. If this option is not set, it is not an active configuration value and is ignored by the server, e.g. all users are allowed in accordance with permissions and other configured options. The group must be defined on Authentication Server before it can be used as a configuration option for Data Management Server.
DMSERVER/SECURE/GRP_DENY	Specifies the name of the group whose members are denied access to Data Management Server. This is an optional setting to provide a convenient way to allow most users and exclude only a few. If this option is not set, it is not an active configuration value and is ignored by the server, e.g. all users are allowed in accordance with permissions and other configured options. The group must be defined on Authentication Server before it can be used as a configuration option for Data Management Server.
DMSERVER/SOAP/CONNS_BACKLOG	Specifies the maximum size of the connection request queue. The backlog is used when the SOAP server receives more connections than it can accept and process within a specific timeframe. The default is 100.
DMSERVER/SOAP/DATA_SVC/IDLE_TIMEOUT	Specifies the number of seconds to allow a DFWSVC process to remain idle before it is terminated. The default setting is 0 indicating that there is no timeout. Negative

Configuration Option	Description
	values are ignored.
DMSERVER/SOAP/DATA_SVC/JOB_COUNT_MAX	Specifies the maximum number of instances of a given service job that can be loaded at the same time. Once this number is reached, Data Management Server will not load any more instances of that service job. Usage: = *count:job_file_name*
DMSERVER/SOAP/DATA_SVC/JOB_COUNT_MIN	Specifies the minimum number of instances of a given service job that must remain loaded. Once this number is reached, Data Management Server will not unload any more instances of that service job. Usage: = *count:job_file_name*
DMSERVER/SOAP/DATA_SVC/MAX_ERRS	Specifies the maximum number of service errors that can occur in a **dfwsvc** process before it is forced to terminate. The default is -1, meaning there is no limit.
DMSERVER/SOAP/DATA_SVC/MAX_NUM	Specifies the maximum number of real-time data services the SOAP server is allowed to run simultaneously. The default is ten. If a new service request would exceed the set limit and queue is not enabled, an error message is displayed. This option applies to the SOAP server, meaning the service requests are coming from a SOAP client. It does not apply to the WLP server or requests coming from a WLP client.
DMSERVER/SOAP/DATA_SVC/MAX_REQUESTS	Specifies the maximum number of service requests a **dfwsvc** process is allowed to handle before it is forced to terminate. The default is -1, meaning this is no limit.
DMSERVER/SOAP/DATA_SVC/MAX_RUNTIME	Specifies the number of seconds to allow a data service to run a job and how long it has to produce a response (output data or an error). If a data service does not produce a response within the configured number of seconds, Data Management Server terminates the corresponding dfwsvc process and sends a SOAP Fault message to the client. The default is zero, meaning no timeout occurs. Negative values are ignored. Note that the timeout count may be off by approximately a second or two due to rounding up of counts less than a second

Configuration Option	Description
	(e.g. 1.5 = 2 secs).
DMSERVER/SOAP/DATA_SVC/PRELOAD	Specifies services and the count for each service that Data Management Server preloads during startup. This can be used with DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL. For more information, see Configure the Server to Pre-load Services.
DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL	Specifies that the Data Management Server should find and preload all services a specific number of times. This includes services found in subdirectories. The number of instances specified must be an integer greater than zero, or the directive is ignored. This can be used with DMSERVER/SOAP/DATA_SVC/PRELOAD.
DMSERVER/SOAP/DATA_SVC/PRELOAD_DURING_RUN	Specifies whether Data Management Server starts a separate thread to preload services and accept SOAP requests at run-time. By default Data Management Server preloads all configured services *before* starting to accept SOAP requests. The same applies if this option is set to ***no***. When this option is set to ***yes***, Data Management Server starts a separate thread to preload all configured services at run-time, while accepting SOAP requests at the same time. If Data Management Server is stopped while the preload thread is still running, that thread will be terminated.
DMSERVER/SOAP/DATA_SVC/QUEUE	Specifies whether to queue real-time data service requests. If set to **yes** and all running dfwsvc processes are busy and Data Management Server is not allowed to start another one to handle a new service request, the request is put in a queue. As soon as a dfwsvc process becomes available, the request is handled. The default is **no** and in the above scenario, an error message is displayed.
DMSERVER/SOAP/IGNORE_NS	Specifies whether or not to ignore namespaces in SOAP requests. The default value is **no**, which means NULL values in input data fields are passed to jobs as empty strings. If set to **yes**, Data Management

Configuration Option	Description
	Server will ignore namespaces in SOAP requests, which allows gSOAP to preserve NULL values when receiving input data instead of converting the values to empty strings.
DMSERVER/SOAP/LISTEN_HOST	Specifies the hostname or IP address to which the SOAP server must bind. A machine running Data Management Server might be available on the network under different hostnames or IP addresses, and different machines on the network may have access to this machine via one hostname or IP address, but not via others. By binding to a particular hostname or IP address, the server only hears requests addressed specifically to that hostname or IP address. A pair of hostnames and IP addresses can be used interchangeably. For example, if this option is set to **localhost**, local clients sending requests to 127.0.0.1 will still be heard. However, requests to public IP address or the hostname of the machine (or requests coming from external clients to the machine) will not be heard. By default, this option is left blank. That means Data Management Server is not bound to any specific hostname or the IP address of the machine and will receive all requests that are coming to the machine, on the correct port.
DMSERVER/SOAP/LISTEN_PORT	Specifies the port on which the SOAP server listens for connections. The default port is 21036.
DMSERVER/SOAP/LOG_PACKETS	Specifies whether to generate the _PACKETS_ log file. If set to **yes**, the log file is generated in the same directory as the input, output, and error SOAP packets log files. For performance reasons, the default value is **no**.
DMSERVER/SOAP/PROC_SVC/MAX_NUM	Specifies the maximum number of real-time process services that Data Management Server is allowed to run simultaneously. The default is ten. If a new service request exceeds this limit, an error message is displayed.
DMSERVER/SOAP/RDWR_TIMEOUT	Specifies the timeout value for read and

Configuration Option	Description
	write operations on a socket. Set a positive value to seconds, a negative value to microseconds, and no timeout to 0. If a non-0 value is set, a timeout occurs if no data can be sent or received within the configured time after the server initiates a send or receive operation over the socket. When a timeout occurs, a SOAP_EOF error is returned. The default is 0 seconds.
DMSERVER/SOAP/RETURN_NULLS	Specifies what the real-time data service returns for null values in output fields. The default value is set at **no**, which means that empty strings are returned. If set to **yes**, NULLs are returned.
DMSERVER/SOAP/SSL	Specifies whether to enable SOAP communication over SSL. The default value is **no**. Set these additional configurations if SSL is configured: • DMSERVER/SOAP/SSL/CA_CERT_FILE • DMSERVER/SOAP/SSL/CA_CERT_PATH • DMSERVER/SOAP/SSL/KEY_FILE • DMSERVER/SOAP/SSL/KEY_PASSWD
DMSERVER/SOAP/SSL/CA_CERT_FILE	Specifies the file where the Certificates Authority stores trusted certificates. If this configuration directive is not needed, comment it out.
DMSERVER/SOAP/SSL/CA_CERT_PATH	Specifies the path to the directory where trusted certificates are stored. If this configuration directive is not needed, comment it out.
DMSERVER/SOAP/SSL/KEY_FILE	Specifies the path to the key file that is required when the SOAP server must authenticate to clients. If this configuration directive is not used, comment it out.
DMSERVER/SOAP/SSL/KEY_PASSWD	Specifies the password for DMSERVER/SOAP/SSL/KEY_FILE. If the key file is not password protected, this configuration should be commented out.

Configuration Option	Description
DMSERVER/SOAP/WSDL	Specifies whether or not to load WSDLs when starting Data Management Server. If job-specific WSDL functionality is needed, set this option to **yes**. This will enable Data Management Server to load existing WSDLs on startup and to recognize jobs that are WSDL-based runSVC requests, that is, if matching WSDLs exist. This will also allow Data Management Server to recognize other WSDL configuration options. The default setting is **no**, which means that existing WSDLs are not loaded, new ones are not generated, no job WSDL-based runSVC requests are recognized, and other WSDL configuration options will not be processed.
DMSERVER/SOAP/WSDL/GEN	Specifies whether or not to allow generation of run-time WSDLs. The default setting is **no**, which means the only WSDLs available to Data Management Server are previously-generated WSDLs that are loaded at startup of Data Management Server. Requests to generate a WSDL or to upload a job with WSDL generation will return errors resulting in the job not uploading. Set the option to **single** to enable generation of a single WSDL per request, e.g. postJob request or genWSDL request for a single file. Set the option to **multiple** to enable generation of multiple WSDLs per request, e.g. genWSDL request for multiple job files or for entire directories of jobs. Note that generating a WSDL can be a time-consuming and resource-intensive process which depends on the parameters of the originating job. Also of note is that a malicious or uninformed user who sends a request to generate WSDLs for all jobs under the root directory can cause a severe degradation in server performance while the WSDLs are being generated; therefore, use the **multiple** option with caution.
DMSERVER/SOAP/WSDL/GEN_ON_GET	Specifies whether or not to allow WSDL generation as part of an HTTP getWSDL request. The default setting is **no**, which means that an error is returned when a WSDL is requested if: 1) it does not already exist, or 2) if it exists but no longer matches

Configuration Option	Description
	the job's **`mod.time`**. If set to **yes** in the situations described above Data Management Server attempts to generate the latest WSDL and if successful, returns the WSDL. This option is activated only when DMSERVER/SOAP/WSDL/GEN is set at **single** or **multiple**. If DMSERVER/SOAP/WSDL/GEN is set at **no**, this option is ignored and WSDLs will not be generated on HTTP getWSDL requests.
DMSERVER/SOAP/WSDL/RUN_IGNORE_MTIME	Specifies whether or not to ignore a difference between WSDL and job **`mod.time`** stamp when a runSVC request comes in. The default setting is **no**, which means that when a client sends a request to run a service based on a job, WSDL and Data Management Server determine that the WSDL no longer matches **`mod.time`** of the service job file, Data Management Server will return a SOAP Fault message '*Job Has Changed*'. The client knows then that the job has changed on Data Management Server and in response, will regenerate the WSDL, obtain it from Data Management Server, and rebuild the client against the latest WSDL. If the option is set to **yes**, Data Management Server will not check whether its WSDL **`mod.time`** matches that of the service job and will pass the received service request to the service process for execution. Note that this is the behavior of service requests based on generic WSDL.
DMSERVER/THREADS/COUNT_MAX	Specifies the number of threads Data Management Server can start. The default value is 1026 threads. If the setting is too low, it is adjusted automatically. There is no setting for an unlimited number of threads. For optimal performance configure the number of threads based on the expected number of parallel clients and requests.
DMSERVER/THREADS/IDLE_MAX	Specifies the number of idle threads Data Management Server can keep. The default is 0; if a thread becomes idle, it is terminated. If it is needed again, it is restarted.
DMSERVER/THREADS/IDLE_TIMEOUT	Specifies the number of microseconds before

Configuration Option	Description
	a thread is flagged as idle after it stops doing work. The default is 0; threads are initially flagged as idle.
DMSERVER/WLP	Specifies if WLP server is enabled. Valid configuration options are **yes** and **no**. The default is **no**. If set to **yes**, WLP Server is started and uses its own listen port. The default is no meaning that WLP server is bypassed during startup of Data Management Server. This means that WLP clients can not connect to Data Management Server but SOAP clients can. The Data Management Server log will contain entries for the status of WLP server.
DMSERVER/WLP/DATA_SVC/MAX_NUM	Specifies the maximum number of real-time data services the WLP server is allowed to run simultaneously. The default is 10. If a new service request exceeds this limit, an error message is displayed. This option applies to the WLP server, meaning the service requests are coming from a WLP client. It does not apply to the SOAP server or requests coming from a SOAP client.
DMSERVER/WLP/LISTEN_HOST	Specifies the hostname or IP address to which the WLP server must bind. By default, this option is left blank. For more information on binding to a hostname or IP address, see DMSERVER/SOAP/LISTEN_HOST.
DMSERVER/WLP/LISTEN_PORT	Specifies the port on which the WLP server listens for connections from WLP clients. If you are running multiple instances of the server on the same machine, each instance must have a unique port configured for it. The default port is 21037.
DMSERVER/WORK_ROOT_PATH	Specifies the root directory under which Data Management Server work and log subdirectories are created. Each time the server starts, a new work directory is created for that instance of the server. The name of this directory contains the server startup date and time, as well as the corresponding process ID. The default directory is *install-path*\var\server_logs.

Code Examples

- Java
- C#

Data Management Server's client library is available in Java and C and can be customized to your environment using the Web Service Definition Language (WSDL) file. The WSDL file contains the descriptions of the available web services. The file, **arch.wsdl**, is installed in the **share** directory of the installation path. You can access the WSDL file via the installation directory path or by using the following URL in a web browser:

```
http://yourserver.yourdomain.com:port/?wsdl
```

In the WSDL file, the value of the **SOAP**:**address** location reflects the local server's hostname and port number. Using an XML editor, you can update the **SOAP:address** location to reflect the hostname and port number of any Data Management Server. One note of caution, please do not edit any other values in **arch**.**wsdl**. For example:

```
<service name="dfx-DMServer-instance-name">
  <documentation>Data Management Server</documentation>
  <port name="DQISService" binding="tns:ArchitectService">
      <SOAP:address
location="http://yourserver.yourdomain.com:21036"/>
  </port>
</service>
```

Java

Use a mapping tool, such as wscompile, to generate stubs and build classes that wrap the Data Management Server interface. In the examples below, using wscompile, the WSDL file is imported and the stubs are generated information from the WSDL file. To use a stub, it must be configured with the service endpoint, or server address.

```
////////////////////////////////////////////////////////

// Imports

////////////////////////////////////////////////////////

import arch.*;

////////////////////////////////////////////////////////

// INITIALIZATION

////////////////////////////////////////////////////////

ArchitectServicePortType_Stub stub;

// get the stub
```

```
stub =(ArchitectServicePortType_Stub)new
DQISService_Impl()).getDQISService();
// optionally set to point to a different end point
stub._setProperty(javax.xml.rpc.Stub.ENDPOINT_ADDRESS_PROPERTY,
"http://MY_SERVER:PORT");

////////////////////////////////////////////////////////
// 1) Get Object List example
////////////////////////////////////////////////////////
String[] res;
res=stub.getObjectList(ObjectType.ARCHSERVICE);

////////////////////////////////////////////////////////
// 2) Post Object example
////////////////////////////////////////////////////////
byte[] myData; ObjectDefinition obj = new ObjectDefinition();
obj.setObjectName("NAME");
obj.setObjectType(ObjectType.fromString("ARCHSERVICE"));
// read the job file in from the h/d
myData = getBytesFromFile(new File(filename));
// post the job to the server
String res=stub.postObject(obj, myData);

////////////////////////////////////////////////////////
// 3) Delete Object
////////////////////////////////////////////////////////
ObjectDefinition obj = new ObjectDefinition();
obj.setObjectName("MYJOB.ddf");
obj.setObjectType(ObjectType.fromString("ARCHSERVICE"));
String res = stub.deleteObject(obj);

////////////////////////////////////////////////////////
// 4) Get Data Service Params
////////////////////////////////////////////////////////
GetArchitectServiceParamResponse resp;
FieldDefinition[] defs;
resp=stub.getArchitectServiceParams("MYJOB.ddf","");
// Get Definitions for Either Input or Output
defs=resp.getInFldDefs();
defs=resp.getOutFldDefs();
//Loop through Defs
defs[i].getFieldName();
defs[i].getFieldType();
defs[i].getFieldLength();

////////////////////////////////////////////////////////
// 5) Execute Data Service
////////////////////////////////////////////////////////
FieldDefinition[] defs;
DataRow[] rows;
String[] row;
GetArchitectServiceResponse resp;
// Fill up the Field Definitions
defs=new FieldDefinition[1];
defs[0] = new FieldDefinition();
defs[0].setFieldName("NAME");
```

```
defs[0].setFieldType(FieldType.STRING);
defs[0].setFieldLength(15);
// Fill up Data matching the definition
rows = new DataRow[3];
row=new String[1];
row[0] ="Test Data";

rows[i] = new DataRow();
rows[i].setValue(row[0]);

resp=stub.executeArchitectService("MYJOB.ddf", defs, rows, "");
// Get the Status, Output Fields and Data returned from the Execute Call
String res = resp.getStatus();
defs=resp.getFieldDefinitions();
rows=resp.getDataRows();
// Output Field Definitions
defs[i].getFieldName();
defs[i].getFieldType();
defs[i].getFieldLength();
// Output Data
row=rows[i].getValue();
res=row[j];

////////////////////////////////////////////////////////
// 6) Run Batch Job
////////////////////////////////////////////////////////
ArchitectVarValueType[] vals;
vals=new ArchitectVarValueType[1];
```

```
vals[0]=new ArchitectVarValueType();
vals[0].setVarName("TESTVAR");
vals[0].setVarValue("TESTVAL");
// Returns JOBID
String res=stub.runArchitectJob("MYJOB.ddf", vals, "");


////////////////////////////////////////////////////////
// 7) Get Job Status
////////////////////////////////////////////////////////
JobStatusDefinition[] defs;
// if you wanted the status for a single job, you would
// pass the jobid returned from runArchitectJob or runProfileJob
defs=stub.getJobStatus("");

ObjectDefinition obj;
obj=defs[i].getJob();
defs[i].getJobid();
defs[i].getStatus();
obj.getObjectName()
obj.getObjectType()

////////////////////////////////////////////////////////
// 8) Get Job Log
////////////////////////////////////////////////////////
GetJobLogResponseType resp;
FileOutputStream fo;
resp=stub.getJobLog(jobId,0);
// write it to a file
```

```
fo = new FileOutputStream (resp.getFileName());

fo.write(resp.getData());

fo.close();


////////////////////////////////////////////////////////

// 9) Terminate Job

////////////////////////////////////////////////////////

String res=stub.terminateJob(jobId);


////////////////////////////////////////////////////////

// 10) Clear Log

////////////////////////////////////////////////////////

String res=stub.deleteJobLog(jobId);
```

C#

Using the DataFlux WSDL file, **arch.wsdl**, import a web reference into your project. This builds the object required to interface with the Data Management Server.

```
////////////////////////////////////////////////////////
// Imports
////////////////////////////////////////////////////////
// Add Web reference using the DataFlux supplied WSDL

////////////////////////////////////////////////////////
// INITIALIZATION
////////////////////////////////////////////////////////
DQISServer.DQISService mService= new DQISServer.DQISService();
mService.Url = "http://MYDISSERVER" + ":" + "PORT";

////////////////////////////////////////////////////////
// 1) Get Object List example
////////////////////////////////////////////////////////
string[] jobs;
jobs=mService.GetObjectList(DQISServer.ObjectType.ARCHSERVICE);

////////////////////////////////////////////////////////
// 2) Post Object example
```

```
///////////////////////////////////////////////////////
DQISServer.ObjectDefinition def = new DQISServer.ObjectDefinition();
def.objectName = "VerifyAddress.ddf";
def.objectType = DQISServer.ObjectType.ARCHSERVICE;

// Grab Bytes from a job file
byte[] data = new byte[short.MaxValue];
FileStream fs = File.Open(@"c:\Develop\SoapUser\VerifyAddress.ddf",
FileMode.Open, FileAccess.Read, FileShare.None);
fs.Read(data,0,data.Length);

DQISServer.SendPostObjectRequestType req= new
DQISServer.SendPostObjectRequestType();
req.@object = def;
req.data = data;

mService.PostObject(req);

///////////////////////////////////////////////////////
// 3) Delete Object
///////////////////////////////////////////////////////
DQISServer.SendDeleteObjectRequestType req = new
DQISServer.SendDeleteObjectRequestType();
DQISServer.ObjectDefinition def = new DQISServer.ObjectDefinition();
def.objectName = "VerifyAddress.ddf";
def.objectType = DQISServer.ObjectType.ARCHSERVICE;

req.job = def;
mService.DeleteObject(req);

///////////////////////////////////////////////////////
// 4) Get Data Service Params
///////////////////////////////////////////////////////
DQISServer.GetArchitectServiceParamResponseType resp;
DQISServer.SendArchitectServiceParamRequestType req;

req=new DQISServer.SendArchitectServiceParamRequestType();
req.serviceName="MYJOB";

resp=mService.GetArchitectServiceParams(req);
string val;
int i;
DQISServer.FieldType field;
// loop through this data
val = resp.inFldDefs[0].fieldName;
i = resp.inFldDefs[0].fieldLength;
field = resp.inFldDefs[0].fieldType;

val = resp.outFldDefs[0].fieldName;
i = resp.outFldDefs[0].fieldLength;
field = resp.outFldDefs[0].fieldType;

///////////////////////////////////////////////////////
// 5) Execute Data Service
///////////////////////////////////////////////////////
```

```
DQISServer.SendArchitectServiceRequestType req = new
DQISServer.SendArchitectServiceRequestType();
DQISServer.GetArchitectServiceResponseType resp;

////////////////////////////////////////////////////////
DQISServer.GetArchitectServiceParamResponseType respParam;
DQISServer.SendArchitectServiceParamRequestType reqParam;
reqParam=new DQISServer.SendArchitectServiceParamRequestType();
reqParam.serviceName="ServiceName";
respParam=mService.GetArchitectServiceParams(reqParam);
////////////////////////////////////////////////////////

DQISServer.FieldDefinition[] defs;
DQISServer.DataRow[] data_rows;
string[] row;

defs=new DQISServer.FieldDefinition[respParam.inFldDefs.Length];
for(int i=0; i < respParam.inFldDefs.Length; i++)
{
     // Fill up the Field Definitions
     defs[i] = new DQISServer.FieldDefinition();
     defs[i].fieldName = respParam.inFldDefs[i].fieldName;
     defs[i].fieldType = respParam.inFldDefs[i].fieldType;
     defs[i].fieldLength = respParam.inFldDefs[i].fieldLength;
}
DataTable table = m_InputDataSet.Tables["Data"]; // externally provided data
// Fill up Data matching the definition
data_rows = new DQISServer.DataRow[Number of Rows];
for(int i=0;i < table.Rows.Count;i++)
{
     System.Data.DataRow myRow = table.Rows[i];
     row=new String[table.Columns.Count];
     for(int c=0;c < table.Columns.Count;c++)
     {
                              row[c] = myRow[c].ToString();
     }
     // Loop and create rows of data to send to the service
     data_rows[i] = new DQISServer.DataRow();
     data_rows[i].value = new string[table.Columns.Count];
     data_rows[i].value = row;
}
req.serviceName = "ServiceName";
req.fieldDefinitions = defs;
req.dataRows = data_rows;
resp=mService.ExecuteArchitectService(req);

////////////////////////////////////////////////////////
// 6) Run Batch Job
////////////////////////////////////////////////////////
DQISServer.SendRunArchitectJobRequest req = new
DQISServer.SendRunArchitectJobRequest();
DQISServer.GetRunArchitectJobResponse resp;

DQISServer.ArchitectVarValueType[] varVal = new
DQISServer.ArchitectVarValueType[1];
```

```
varVal[0] = new DQISServer.ArchitectVarValueType();
varVal[0].varName = "TESTVAR";
varVal[0].varValue = "TESTVAL";

req.job = "JOB_NAME";
req.varValue = varVal;

resp = mService.RunArchitectJob(req);

string jobid = resp.jobId;

////////////////////////////////////////////////////////
// 7) Get Job Status
////////////////////////////////////////////////////////
DQISServer.SendJobStatusRequestType req = new
DQISServer.SendJobStatusRequestType();
DQISServer.JobStatusDefinition[] resp;
req.jobId = "";

resp = mService.GetJobStatus(req);
DQISServer.ObjectDefinition def = resp[0].job;
string jobid = resp[0].jobid;
string jobstatus = resp[0].status;

////////////////////////////////////////////////////////
// 8) Get Job Log
////////////////////////////////////////////////////////
DQISServer.SendJobLogRequestType req = new
DQISServer.SendJobLogRequestType();
DQISServer.GetJobLogResponseType resp;
req.jobId = "SOMEJOBID";

resp = mService.GetJobLog(req);
string fileName = resp.fileName;
byte []data = resp.data;

////////////////////////////////////////////////////////
// 9) Terminate Job
////////////////////////////////////////////////////////
DQISServer.SendTerminateJobRequestType req = new
DQISServer.SendTerminateJobRequestType();
DQISServer.GetTerminateJobResponseType resp;
req.jobId = "SOMEJOBID";

resp = mService.TerminateJob(req);
string fileName = resp.status;

////////////////////////////////////////////////////////
// 10) Clear Log
////////////////////////////////////////////////////////
DQISServer.SendDeleteJobLogRequestType req = new
DQISServer.SendDeleteJobLogRequestType();
DQISServer.GetDeleteJobLogResponseType resp;
req.jobId = "SOMEJOBID";
```

```
resp = mService.DeleteJobLog(req);
string fileName = resp.status;
```

Legal Notices

DataFlux Legal Statements

DataFlux Solutions and Accelerators Legal Statements

DataFlux Legal Statements

Apache Portable Runtime License Disclosure

Copyright © 2008 DataFlux Corporation LLC, Cary, NC USA.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache/Xerces Copyright Disclosure

The Apache Software License, Version 3.1

Copyright © 1999-2003 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

   "This product includes software developed by the Apache Software Foundation (http://www.apache.org)."

   Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc., http://www.ibm.com. For more information on the Apache Software Foundation, please see http://www.apache.org.

Boost Software License Disclosure

Boost Software License - Version 1.0 - August 17, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

DataDirect Copyright Disclosure

Portions of this software are copyrighted by DataDirect Technologies Corp., 1991 - 2008.

Expat Copyright Disclosure

Part of the software embedded in this product is Expat software.

Copyright © 1998, 1999, 2000 Thai Open Source Software Center Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

gSOAP Copyright Disclosure

Part of the software embedded in this product is gSOAP software.

Portions created by gSOAP are Copyright © 2001-2004 Robert A. van Engelen, Genivia inc. All Rights Reserved.

THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY GENIVIA INC AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

IBM Copyright Disclosure

ICU License - ICU 1.8.1 and later [used in DataFlux Data Management Platform]

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1995-2005 International Business Machines Corporation and others. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Microsoft Copyright Disclosure

Microsoft®, Windows, NT, SQL Server, and Access, are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle Copyright Disclosure

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates.

PCRE Copyright Disclosure

A modified version of the open source software PCRE library package, written by Philip Hazel and copyrighted by the University of Cambridge, England, has been used by DataFlux for regular expression support. More information on this library can be found at: ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/.

Copyright © 1997-2005 University of Cambridge. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Red Hat Copyright Disclosure

Red Hat® Enterprise Linux®, and Red Hat Fedora™ are registered trademarks of Red Hat, Inc. in the United States and other countries.

SAS Copyright Disclosure

Portions of this software and documentation are copyrighted by SAS® Institute Inc., Cary, NC, USA, 2009. All Rights Reserved.

SQLite Copyright Disclosure

The original author of SQLite has dedicated the code to the public domain. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

Sun Microsystems Copyright Disclosure

Java™ is a trademark of Sun Microsystems, Inc. in the U.S. or other countries.

USPS Copyright Disclosure

National ZIP®, ZIP+4®, Delivery Point Barcode Information, DPV, RDI, and NCOA^{Link}®. © United States Postal Service 2005. ZIP Code® and ZIP+4® are registered trademarks of the U.S. Postal Service.

DataFlux is a non-exclusive interface distributor of the United States Postal Service and holds a non-exclusive license from the United States Postal Service to publish and sell USPS CASS, DPV, and RDI information. This information is confidential and proprietary to the United States Postal Service. The price of these products is neither established, controlled, or approved by the United States Postal Service.

VMware

VMware® virtual environment provided those products faithfully replicate the native hardware and provided the native hardware is one supported in the applicable DataFlux product documentation. All DataFlux technical support is provided under the terms of a written license agreement signed by the DataFlux customer.

The VMware virtual environment may affect certain functions in DataFlux products (for example, sizing and recommendations), and it may not be possible to fix all problems.

If DataFlux believes the virtualization layer is the root cause of an incident; the customer will be directed to contact the appropriate VMware support provider to resolve the VMware issue and DataFlux shall have no further obligation for the issue.

Solutions and Accelerators Legal Statements

Components of DataFlux Solutions and Accelerators may be licensed from other organizations or open source foundations.

Apache

This product may contain software technology licensed from Apache.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at: http://www.apache.org/licenses/LICENSE-2.0.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Creative Commons Attribution

This product may include icons created by Mark James http://www.famfamfam.com/lab/icons/silk/ and licensed under a Creative Commons Attribution 2.5 License: http://creativecommons.org/licenses/by/2.5/.

Degrafa

This product may include software technology from Degrafa (Declarative Graphics Framework) licensed under the MIT License a copy of which can be found here: http://www.opensource.org/licenses/mit-license.php.

Copyright © 2008-2010 Degrafa. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Google Web Toolkit

This product may include Google Web Toolkit software developed by Google and licensed under the Apache License 2.0.

JDOM Project

This product may include software developed by the JDOM Project (http://www.jdom.org/).

OpenSymphony

This product may include software technology from OpenSymphony. A copy of this license can be found here: http://www.opensymphony.com/osworkflow/license.action. It is derived from and fully compatible with the Apache license that can be found here: http://www.apache.org/licenses/.

Sun Microsystems

This product may include software copyrighted by Sun Microsystems, jaxrpc.jar and saaj.jar, whose use and distribution is subject to the Sun Binary code license.

This product may include Java Software technologies developed by Sun Microsystems,Inc. and licensed to Doug Lea.

The Java Software technologies are copyright © 1994-2000 Sun Microsystems, Inc. All rights reserved.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. DATAFLUX CORPORATION LLC, SUN MICROSYSTEMS, INC. AND THEIR RESPECTIVE LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN MICROSYSTEMS, INC. OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN MICROSYSTEMS, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Java Toolkit

This product includes the Web Services Description Language for Java Toolkit 1.5.1 (WSDL4J). The WSDL4J binary code is located in the file wsdl4j.jar.

Use of WSDL4J is governed by the terms and conditions of the Common Public License Version 1.0 (CPL). A copy of the CPL can be found here at http://www.opensource.org/licenses/cpl1.0.php.

Glossary

A

ACE

An access control entry (ACE) is an item in an access control list used to administer object and user privileges such as read, write, and execute.

ACL

Access control lists (ACLs) are used to secure access to individual Data Management Server objects.

API

An application programming interface (API) is a set of routines, data structures, object classes and/or protocols provided by libraries and/or operating system services in order to support the building of applications.

D

DAC

A data access component (DAC) allows software to communicate with databases and manipulate data.

dfwfproc

A process handled by Data Management Server that runs process services, batch jobs, and profile jobs

dfwsvc

A Data Management Server process that runs real time services.

DPV

Delivery Point Validation (DPV) is a USPS database that checks the validity of residential and commercial addresses.

DSN

A data source name (DSN) contains connection information, such as user name and password, to connect through a database through an ODBC driver.

L

LACS

Locatable Address Conversion System (LACS) is used updated mailing addresses when a street is renamed or the address is updated for 911, usually by changing a rural route format to an urban/city format.

M

MMC

The Microsoft Management Console (MMC) is an interface new to the Microsoft Windows 2000 platform which combines several administrative tools into one configurable interface.

O

ODBC

Open Database Connectivity (ODBC) is an open standard application programming interface (API) for accessing databases.

OpenSSL

The open source implementation of SSL. See SSL.

P

PID

Process ID; a number used to uniquely identify a process.

Q

QAS

Quick Address Software (QAS) is used to verify and standardize US addresses at the point of entry. Verification is based on the latest USPS address data file.

QKB

The Quality Knowledge Base (QKB) is a collection of files and configuration settings that contain all DataFlux data management algorithms. The QKB is directly editable using DataFlux Data Management Studio.

R

RDI

Residential Delivery Indicator (RDI) identifies addresses as residential or commercial.

S

SERP

The Software Evaluation and Recognition Program (SERP) is a program the Canadian Post administers to certify address verification software.

SOA

Service Oriented Architecture (SOA) enables systems to communicate with the master customer reference database to request or update information.

SOAP

Simple Object Access Protocol (SOAP) is a Web service protocol used to encode requests and responses to be sent over a network. This XML-based protocol is platform independent and can be used with a variety of internet protocols.

SSL

Secure Sockets Layer; security protocol to enable Web sites to pass sensitive information securely in an encrypted format.

U

USPS

The United States Postal Service (USPS) provides postal services in the United States. The USPS offers address verification and standardization tools.

W

WSDL

Web Services Definition Language: an XML-based language that provides a model for describing Web services.