THE POWER TO KNOW.

DataFlux® Data Management Server 2.5

Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2013. *DataFlux® Data Management Server 2.5: Administrator's Guide*. Cary, NC: SAS Institute Inc.

DataFlux® Data Management Server 2.5: Administrator's Guide

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

Printing 2, December 2013

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit **support.sas.com/bookstore** or call 1-800-727-3228.

Contents

What's New in DataFlux Data Management Server

Overview

The primary enhancements for DataFlux Data Management Server include the following:

- new server configuration and security
- SAS Metadata Server configured by default for security
- SAS Metadata Server provides configuration options at server start
- collect job status information with the SAS Job Monitor
- record log entries that contain alternate encodings
- terminate real-time services
- support for SAS 9.4 data

New Server Configuration and Security

The DataFlux Data Management Server is now delivered in a single edition, which replaces the former Standard and Enterprise editions. The new configuration provides the capabilities of the former Enterprise edition. Also, the DataFlux Secure software is now installed by default, in a disabled state, on all instances of DataFlux Data Management Server (except where prohibited by export restrictions). You can enable enhanced encryption, including the 256-bit private keys of the Advanced Encryption Standard (AES). You can also enable the DataFlux Data Management Server to accept SOAP connections only from clients that use Transport Security Layer (TSL) or Single Sockets Layer (SSL) protection (HTTPS addresses).

SAS Metadata Server Configured by Default

The DataFlux Data Management Server is now configured by default to use a SAS Metadata Server for authorization and authentication. The SAS Metadata Server provides exactly the same support that was previously provided by the DataFlux Authentication Server. If your site uses a DataFlux Authentication Server, you can

configure your DataFlux Data Management Server to use it by editing a single configuration file.

SAS Metadata Server Provides Configuration Options at Server Start

When you use a SAS Metadata Server for security and when you start the DataFlux Data Management Server, you download several configuration options from the DataFlux Data Management Server's metadata definition. You can change these options in SAS Management Console. The downloaded options should not be set in local configuration files.

Collect Job Status Information with the SAS Job Monitor

If your site uses the SAS Environment Manager, then you can use the SAS Job Monitor to collect status information about the jobs that run on the DataFlux Data Management Server. You collect status information at the end of job execution by default. You can choose to collect information at specified time periods during job execution.

Record Log Entries That Contain Alternate Encodings

You can set the configuration option BASE/JOB_LOG_ENCODING to specify a non-default character encoding for batch job log entries.

Terminate Real-Time Services

SOAP clients that request job runs can now terminate those jobs by name. You use SOAP commands to assign a job ID and then unload the process on the server that ran the job. Unloading the process terminates any real-time services that failed to terminate normally.

Support for SAS 9.4 Data

Starting with the first maintenance release, the DataFlux Data Management Server 2.5 can now access SAS 9.4 data files without setting the configuration option EXTENDOBSCOUNTER = NO. The default value EXTENDOBSCOUNTER = YES enables access to data files that contain a much greater number of observations.

Accessibility

The DataFlux Data Management Platform software includes features that improve usability for the disabled. The usability 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, please send an e-mail to techsupport@sas.com.

Recommended Reading

- DataFlux Data Management Server Help
- DataFlux Data Management Studio User's Guide
- DataFlux Data Management Studio Installation and Configuration Guide
- SAS Visual Process Orchestration Server Administrator's Guide
- SAS Intelligence Platform: System Administration Guide
- SAS Intelligence Platform: Security Administration Guide
- SAS Federation Server Administrator's Guide
- DataFlux Authentication Server Administrator's Guide
- DataFlux Web Studio User's Guide
- DataFlux Web Studio Installation and Configuration Guide

For a complete list of SAS books, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Book Sales Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Chapter 1

Introducing the DataFlux Data Management Server

Introduction to the DataFlux Data Management Server

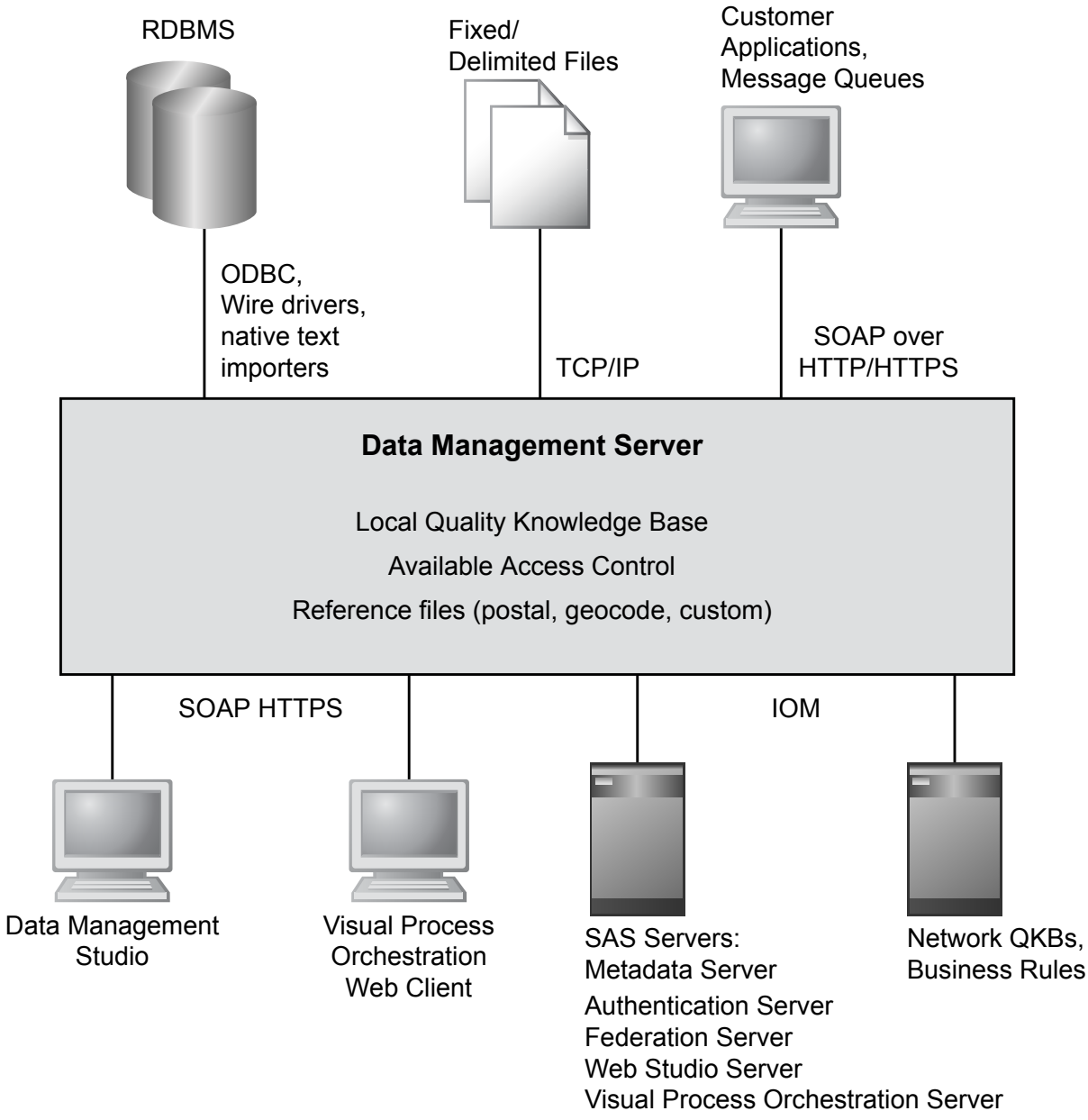
Overview

The DataFlux Data Management Server provides consistent, accurate, and reliable access to data across a network by integrating real-time data quality, data integration, and data governance routines. With DataFlux 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 your enterprise. The server implements business rules that you create in DataFlux Data Management Studio, in both batch and real-time environments. DataFlux Data Management Server enables pervasive data quality, data integration, process orchestration, and master data management (MDM) throughout your enterprise.

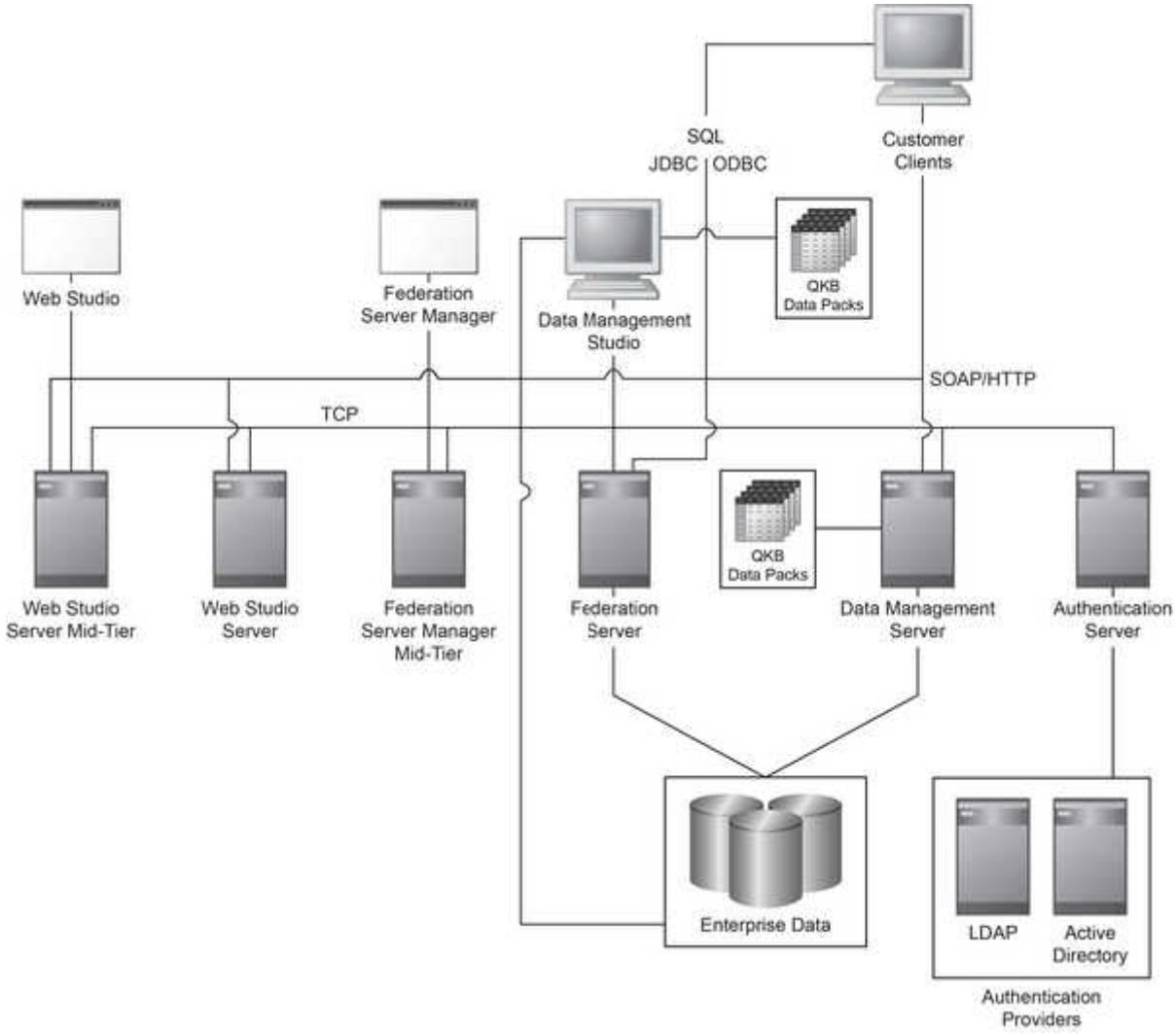
The Data Management Server provides a service-oriented architecture (SOA) application server that enables you to execute batch or profile jobs on a server-based platform, in Windows, Linux, or UNIX. By processing batch and profile jobs where the data resides, you avoid network bottlenecks and take advantage of performance features available with higher-performance computers.

In addition, the Data Management Server executes real-time data services and real-time process services. These services can be invoked by any web service application, such as SAP, Siebel, Tibco, or Oracle. You can convert your existing batch jobs to real-time services, to reuse the business logic that you developed for data migration or to load a data warehouse. You can apply your real-time services at the point of data entry to ensure consistent, accurate, and reliable data across your enterprise.

The following diagram shows how DataFlux Data Management Server connects to other servers and clients:



The following diagram shows how the DataFlux Data Management Server connects into enterprise software solutions that are based on the DataFlux Data Management Platform:



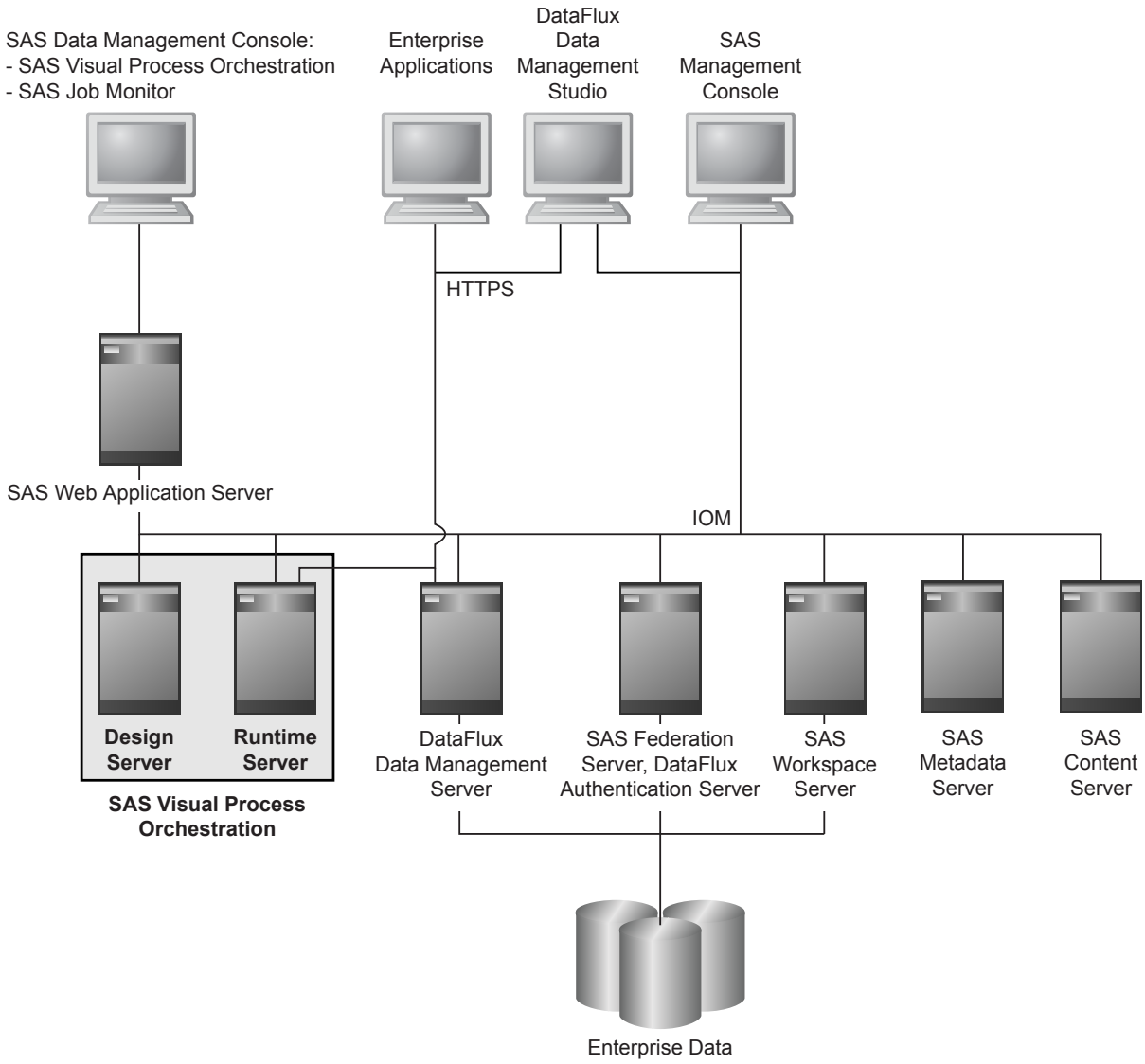
Also included with Data Management Server is the ability to make Application Programming Interface (API) calls to the same core data quality engine. Discrete API calls are available through native programmatic interfaces for data parsing, standardization, match key generation, address verification, geocoding,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

DataFlux Data Management Studio is the development, test, and administration client for the DataFlux Data Management Server. DataFlux Data Management Studio enables you to create, test, and upload batch jobs, profile jobs, real-time data services, and real-time process services. Production jobs can be run by individuals, clients, or by your scheduling application. Your clients and web applications can access DataFlux Data Management Server through a SOAP interface. The Data Management Server also supports a WSDL client interface (Web Services Description Language).

Security on the Data Management Server is implemented through external authentication and internal authorization. External security services are provided either by a SAS Metadata Server or a DataFlux Authentication Server. Both of the security servers authenticate users using network authentication providers. The security servers also maintain a database of users and groups. The DataFlux Data Management Server applies group membership information to its internal access control lists for data, jobs, commands, and services. Additional broad-brush security features grant or deny server access based on IP address and on membership in ALLOW, DENY, or administrative groups. Encryption is applied to all TCP/IP network communication and all stored

passwords. You can configure encryption to use private keys up to 256 bits in length. You can also configure SSL to protect client connections using HTTPS addresses.

The Data Management Server is provided as part of an increasing number of enterprise solutions, including SAS MDM and SAS Visual Process Orchestration. The following diagram shows how the DataFlux Data Management Server provides a job execution environment and an external client interface for SAS Visual Process Orchestration.



How It Works

The DataFlux Data Management Server is responsible not only for sending and receiving SOAP requests, but also for monitoring the progress of all registered data management services. Job status information is available in DataFlux Data Management Studio, and, when configured, in the Job Monitor add-in the SAS Environment Manager software.

On the Data Management Server, SOAP and WLP (web application logic) servers listen on separate ports. When the server receives a job run request, the server authenticates, authorizes, and sends the request to a threaded process. The process runs and executes the real-time data service, real-time process service, batch job, or profile job. When the job is complete, the server sends data to the client and the process is assigned to the next job run request.

You can preload processes, spawn new processes, and enqueue job run requests as needed to customize server performance for the dynamics of your enterprise. The use of separate processes for job run requests enables robust error recovery and effective distribution of processing across multiple CPUs.

The Data Management Server handles the following processes:

- Client queries the 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 requested input/output fields for a specified service.
- Pass data and macros to a service, run the service, and receive output data and macros in return. When the server receives a service request, it identifies an idle service, sends data to the idle service, and listens for additional 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

Overview

The user interface for DataFlux Data Management Server is provided by DataFlux Data Management Studio. To display the interface, open the **Administration** riser bar and click the **DataFlux Data Management Servers** riser bar. DataFlux Data Management Studio then displays a tree view of your Data Management Servers in the left-hand navigation pane. The right-hand information pane displays a list of server names.

About 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 DataFlux 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 DataFlux Data Management Server, so that you can connect to it.

Import
: Enables you to import items from a repository.

Export
: Enables you to export the selected object to a repository.

Edit
: Enables you to export the selected object to a repository.

Expand
: Enables 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.

About the Information Pane

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

New
: Enables you to register a new DataFlux Data Management Server.

Edit
: Enables you to import items from a repository.

Delete
: Enables you to export the selected object to a repository.

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

Introducing the Directories

The following table lists the directories that are created when you install Data Management Server.

Directory	Description
\bin	Contains the executable files for this platform.
\data	Contains files that include data information that is specific to this installation.
\data\install	Contains a collection of files pertinent to installation such as templates and scripts.
\doc	Contains the documentation that is installed with the server.
\etc	Contains the configuration and license files.
\lib	Contains the library files for this platform.
\etc\dftkdsn	Contains the non-ODBC data connection configurations.
\etc\dsn	Contains the saved credential files for each data source name (DSN).

Directory	Description
\etc\license	By default, the location where the license files reside. The path to the license file is located in the etc\app.cfg file.
\etc\macros	Contains the .cfg files, which specify the macro key and value pairs. All files in this directory are loaded in alphabetical order.
\etc\repositories	Contains the sample repository configuration file, server.rcf. The repository configuration file defines the location of the repository file that is used by the server to run profile jobs.
\etc\security	Contains files that specify server commands and permissions for specific users and groups.
\share	Contains message files that are needed by the software. If the files are removed, the software will fail to run. The directory also contains a sample copy of the WSDL file, which is used by the DataFlux Data Management Server.
\var	Contains the log files from the running of the DataFlux Data Management Server as well as job-specific logs.
\var\repositories	Contains the sample repository file, server.rps.

Chapter 2

Configuring the DataFlux Data Management Server

Configure Additional Software

Overview

You can add data cleansing, data quality, and address verification applications to your Data Management Server so that job nodes can access the applications on the local host. These applications are available on the SAS support site, in the downloads and hot fixes section. See **`http://support.sas.com/demosdownloads`**.

You can customize applications such as dfIntelliserver, Quality Knowledge Bases (QKB), Accelerators, and DataPacks to meet the needs of your service-oriented architecture.

For information about installing dfIntelliserver, QKBs, and Accelerators, see the relevant software installation documentation. For information about installing and configuring the DataPacks for address verification, including USPS, Canada Post, and Geocode, see the *DataFlux Data Management Studio Installation and Configuration Guide*.

Address Update

About Address Update

The DataFlux Address Update add-on enables you to use the United States Postal Service (USPS) NCOALink® system to identify and update address information in customer records. For businesses and organizations with very large North America-

based customer information databases, this essential feature maintains accurate and up-to-date address information for location-based marketing and direct mail marketing.

Address update jobs can be imported from DataFlux Data Management Studio to a DataFlux Data Management Server, where the jobs are executed. One approach is to run test jobs and small jobs on the DataFlux Data Management Studio client workstation, and to upload larger jobs to the DataFlux Data Management Server. Using this approach, both DataFlux Data Management Studio and DataFlux Data Management Server must be identically configured to execute address update jobs and reports.

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

Install Address Update

Follow these steps to install Address Update:

1. Before you install Address Update on your DataFlux Data Management Server, install, configure, and test Address Update on an instance of DataFlux Data Management Studio. When your configuration is established on DataFlux Data Management Studio, you can replicate that configuration on your DataFlux Data Management Server.
2. Install on the Data Management Server the same Quality Knowledge Base that is used on DataFlux Data Management Studio. Replicate all customizations.
3. Download and run the Address Update installer on the host of the DataFlux Data Management Server.
4. Install NCOALINK Data from the United States Postal Service (USPS). Follow the instructions in the *Address Update Add-On to DataFlux Data Management Studio Quick Start Guide* that is provided with the Address Update installer.
5. Install USPS test data (CASS, DPV, and LACS), as described in the *Address Update Add-On to DataFlux Data Management Studio Quick Start Guide*.

Configure DataFlux Data Management Server for Address Update

Follow these steps to configure Address Update:

1. If necessary, stop the DataFlux Data Management Server.
2. Open the configuration file ***install-path*/etc/macros/ncoa.cfg**.
3. Set the value of the option NCOA/DVDPATH to the installation path of the USPS NCOALink data.
4. Set the value of the option NCOA/QKBPATH to be the installation path of the Quality Knowledge Base.
5. Set the value of the option NCOA/USPSPATH to be the installation path of the USPS Address verification data.
6. Review the default value of the option NCOA/DFAV_CACHE_SIZE. This option specifies the size of the cache that is used for address verification. The range of valid values is 0–100, and 0 is the default. Increasing the value increases the amount of memory used and increases the performance of address verification.

7. Review the default value of the option NCOA/DFAV_PRELOAD. This option specifies the states and categories of addresses that you preload, to enhance the performance of address verification. Valid values for this option are defined as follows:

   "."
   No preload. This is the default value.

   "ALL"
   Preload all states.

   "MIL"
   Preload military addresses only.

   "CONUSA"
   Preload the 48 contiguous states.

   "TX FL"
   Preload Texas and Florida, or any list of two-digit state names.

8. Save and close the configuration file ncoa.cfg.
9. Open the configuration file ***install-path*/etc/app.cfg**.
10. Update the values of the following options according to the app.cfg file in Data Management Studio:
    - NCOA/REPOSDSN specifies the DSN connection for the address update repository.
    - NCOA/REPOSPREFIX specifies the table prefix for the tables in this repository, if a prefix has been specified.
    - NCOA/REPOSTYPE specifies the type of repository. Valid values for this option are defined as follows:

      0
      No type specified. The DataFlux Data Access Component attempts to determine the repository type from the connection string.

      1
      Specifies the repository type ODBC DSN.

      2
      Specifies the repository type Custom DSN.

Configure Jobs to Use Address Update

After you configure DataFlux Data Management Server to use Address Update, configure Data Management Server jobs to use Address Update. The following steps are described in detail in the DataFlux Data Management Studio Help topic entitled *Online Help, Using the Address Update Add-On with DataFlux Data Management Server*:

1. Create a separate Processing Acknowledgment Form (PAF) for the DataFlux Data Management Server if Data Management Studio and DataFlux Data Management Server are running on different operating systems.
2. Enable jobs on Data Management Server to access an address update repository.
3. Configure a DSN on the DataFlux Data Management Server that is identical to the DSN defined in the NCOA/REPOSDSN option in the app.cfg file. Users need to save credentials for this DSN.
4. Import your Address Update Lookup jobs from DataFlux Data Management Studio to the Batch Jobs folder on the DataFlux Data Management Server.

At this point, you are ready to run your Address Update Lookup jobs.

Configure the Quality Knowledge Base

If you add a QKB to your DataFlux Data Management Server, make sure that it is the same QKB that you installed on DataFlux Data Management Studio.

To specify the location of the QKB, open the configuration file ***install-path*/etc/app.cfg**. For the QKB variable, remove the comment character and replace **PATH** with the full path to the QKB.

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

Configure DataPacks

If you download DataPacks, open ***install-path*/etc/app.cfg**, remove comment characters, and update variable values as follows.

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 DataFlux Data Management Server.

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 indicated by the term ***install-path***.

Table 2.1 *Directory Permissions for 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 DataFlux Data Management Server, Backup Administrator	Read, List Folder Contents

Table 2.2 *Directory Permissions for UNIX and Linux*

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 DataFlux Data Management Server; Backup Administrator	Read, Execute

Note: TMPDIR might 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 DataFlux Data Management Server

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

1. In DataFlux Data Management Studio, click the **DataFlux Data Management Servers** riser bar.
2. Click **New DataFlux Data Management Server** on the toolbar. The **Management Server** dialog box appears.
3. In the **Management Server** dialog box,
   a. Specify a name for the server in the **Name** field.
   b. Enter a description for the server.
   c. Enter the server host name in the **Server** field.
   d. Port **21036** is the default port number. Keep the default port number unless it is already in use on that host.
   e. Enter the **Domain Name** for the associated Authentication Server.
4. Click **Test Connection** to verify that you can connect to the server. Click **OK** to close the Test dialog box.
5. Click **OK** to close the DataFlux Data Management Server dialog box.

The new Data Management Server appears in the left navigation pane.

Configure DataFlux Data Management Server to Run Studio Jobs and Services

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

To run a new job or service, the configuration of the DataFlux Data Management Server needs to replicate the configuration of the Studio client. Certain job 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 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 configuration information refers to the Studio client, but the configuration process also needs to be applied to the DataFlux Data Management Server.

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

- 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 *DataFlux Data Management Studio User's Guide*.

- Set configuration options in the DataFlux Data Management Server configuration file dmserver.cfg. For information about these configuration options, see "Configuration Options Reference for dmserver.cfg".

In addition to transferring the Studio configuration to the DataFlux Data Management Server, you also need to consider the application of access controls to specify the users who will be permitted to run jobs and real-time services.

Migrate Security after Software Upgrades

Security options and objects need to be migrated as part of the process of upgrading to a new release of the DataFlux Data Management Server, the SAS Metadata Server, and the Authentication Server. Do not remove or uninstall the previous version until after you migrate your security settings and objects.

Begin by migrating your SAS Metadata Server (see the *SAS Intelligence Platform: Migration Guide*) or DataFlux Authentication Server (see the *DataFlux Authentication Server: Administrator's Guide*.)

When your authentication and authorization server is migrated, you are ready to migrate your jobs, access control lists, and command permissions to your newly installed DataFlux Data Management Server. Follow these steps:

1. Copy your jobs and services from the previous version of the server to the new version.
2. Transfer the security configuration options in dmserver.cfg from the old version to the new version.
3. Import profile jobs into the new repository.
4. Create user and group command permissions to match those of the previous version.
5. Create ACL permissions for jobs to match those of the previous version.
6. Test the new server.
7. Uninstall the old server if it is no longer in use.

Chapter 3

Managing Security

Security Overview

The Data Management Server 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 minimize risk, maximize utilization of the technology, and expedite deployment.

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 DataFlux 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 SAS Metadata Server or DataFlux Authentication Server** uses either a SAS Metadata Server or a DataFlux Authentication Server to authenticate user credentials and provide group membership information. This mode enables you to use a single set of user and group definitions for your entire enterprise.
- **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. These features are provided by the DataFlux Secure software, which is installed by default in a disabled state. For more information about DataFlux Secure, see "Configure SSL and AES"and the *DataFlux Secure Administrator's Guide*.

When security is not enabled, you cannot run jobs that request authentication, and you cannot run jobs that access a SAS Federation Server.

All data sources (DSNs) needed by jobs and services must be defined on the Data Management Server.

When you upgrade to a new release of the DataFlux Data Management Server, you must manually migrate your security settings from the previous release to the new release. To migrate your security settings, see "Migrate Security after Software Upgrades".

About Authentication

Authentication is the process of confirming the identity of users. When authentication is enabled on a DataFlux Data Management Server, the server requires a connection to a DataFlux Authentication Server or a SAS Metadata Server.

The authentication process begins when the DataFlux Data Management Server receives a connection request from a client. The DataFlux Data Management Server passes the user's credentials to the DataFlux Authentication Server or SAS Metadata Server. The other server then attempts to authenticate the user by submitting the user's credentials to an authentication provider in the network domain that is specified in the credentials. After it receives a response from the authentication provider, the Authentication Server

or SAS Metadata Server notifies the DataFlux Data Management Server of the result of the authentication attempt. Successful authentication enables the DataFlux Data Management Server to begin the authorization process.

Authentication is enabled on the DataFlux Data Management Server with the configuration option DMSERVER/SECURE, in the configuration file dmserver.cfg.

For information about configuring authentication providers and defining users and groups, see the following documents:

- *SAS Intelligence Platform: System Administration Guide*
- *SAS Intelligence Platform: Security Administration Guide*
- *DataFlux Data Management Server: Administrator's Guide*

About Authorization

Overview

The authorization process applies access controls to authenticated users. After a user successfully authenticates, the DataFlux Data Management Server queries the SAS Metadata Server or DataFlux Authentication Server for the group membership information of the authenticated user. The DataFlux Data Management Server applies the group membership information to its locally defined authorizations to allow or deny the user access to the requested object or command.

Note: Authorizations that might be defined on the SAS Metadata Server are not applied to the Data Management Server's authorization process.

Authorizations in the form of access control lists are defined on the DataFlux Data Management Server using the Administration riser in DataFlux Data Management Studio. You can define additional enterprise-level authorizations by setting configuration options. You can allow or deny access to the Data Management Server by IP address. You can also enable ALLOW and DENY groups. If the requesting user is a member, direct or indirect, of the ALLOW or DENY group, then full access is granted or denied, and no further authentication takes place.

Group and User Authorization Checks

The Data Management Server checks a user's authorizations in the following sequence:

1. membership in the administrators group
2. the DENY group, if it is configured for use
3. the ALLOW group, if it is configured for use
4. command permissions and access control lists

Note: All groups must be created on the SAS Metadata Server or DataFlux Authentication Server *before* they can be authorized access on the DataFlux Data Management Server.

If the requesting user is a member of the administrators group, the DENY group, or the ALLOW group, then access is granted or denied and no further authorization takes place. Members of the administrators group and the ALLOW group are granted access to all DataFlux Data Management Server commands and objects.

After group memberships are compared against the access controls lists, the Data Management Server determines whether the following command permissions are set for the user:

- If, for a given command or object, the user has *deny* set, then the user is denied access. If an ACL exists, it is not checked.
- If the user has *inherit* set, authorization checks proceed to group permissions.
- If the user has *allow* set, and if the request does not apply to a specific object, then the user is granted access. If the request does apply to a specific object, then the server checks the object's ACL.

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 can be determined from the groups that are one step from the user, then the DataFlux Data Management Server will not look further. When the server looks at a set of groups that are the same distance from the user, if any group has the DENY permission, then the user is denied access. Otherwise, if any group has the ALLOW permission, then if there is an ACL to check, the authorization process moves to the ACL. If there is no ACL at this point, then the user receives access. If permissions are not set for any group, or the permission is set to INHERIT, then the authorization checks move to the set of groups one step farther from the user.

If access rights cannot be determined after going through the groups to which the user is a member, then the next group whose permissions are checked is the USERS group. All users that have definitions on the SAS Metadata Server or the DataFlux Authentication Server belong to the USERS group. Administrators can set command permissions for the USERS group and use that group in ACLs in the same manner as any other group.

If access rights have not been determined, based on command permissions, the last step in the authorization process is to check whether permissions are set for the PUBLIC group. The PUBLIC group includes all users who are not registered on the SAS Metadata Server or the DataFlux Authentication Server. If the permission is ALLOW and is there is an ACL to check, then the authorization check moves to the ACL. Otherwise, the user is granted access. If the permission is DENY, INHERITY, or is not set, then the user is denied access.

If neither the user, nor the user’s groups, the USERS group, or the PUBLIC group have permission set, then the DataFlux Data Management Server denies access without checking the ACL. This means that the DataFlux Data Management Server requires a specific command permission before the Data Management Server will look at the ACL of an individual object.

ACL Authorization Checks

Authorization checks of ACLs begin by determining if the user is the owner of the object. If the user is the owner, then the user is granted access to the object. If the object is owned by a group, the user must be a direct or indirect member of that group to be granted access to the object.

Next, the authorization check search the access control entries (ACEs). If ALLOW or DENY permissions are not found for the requesting user, then the ACEs are checked for groups of which the user is a member.

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

Configure a SAS Metadata Server for Security

Overview

When you install a Data Management Server, it is configured by default to use a SAS Metadata Server for authentication and authorization. If your site uses a DataFlux Authentication Server instead of a SAS Metadata Server, then see "Configure a DataFlux Authentication Server for Security".

Basic Configuration Occurs during Installation

When you install a Data Management Server, the SAS Deployment Wizard sets the value of the configuration option BASE/AUTH_SERVER_LOC to specify the network name and port of the SAS Metadata Server. After installation, the file ***install-path*/etc/app.cfg** contains an entry that is similar to this example:

```
base/auth_server_loc=iom://Orion.us.southeast.omr.com:8561
```

Note: 8651 is the default port number for the SAS Metadata Server. Always use this port number unless it is already in use on the host of the SAS Metadata Server.

The SAS Deployment Wizard also creates a metadata definition for the DataFlux Data Management Server on the SAS Metadata Server. After installation, you can see and control the DataFlux Data Management Server in SAS Management Console or in a newer administrative client.

Manage Server Configuration Options That Are Set from Metadata

When you use a SAS Metadata Server for security, you download the values of the following configuration options when you start the DataFlux Data Management Server: DMSERVER/SOAP/SSL, DMSERVER/SOAP/LISTEN PORT, and DMSERVER/SECURE.

The Data Management Server uses the value of DMSERVER/NAME to query its own metadata definition on the SAS Metadata Server. If the name is valid and if the metadata definition can be accessed, then the DataFlux Data Management Server sets the local values from the supplied metadata.

To access the metadata definition, the process owner of the DataFlux Data Management Server must have a user definition on the SAS Metadata Server. Another method of enabling access is to specify Read access to the metadata definition for the PUBLIC group.

If the metadata definition cannot be accessed by the specified name, or if the name is valid and if access is denied, then the DataFlux Data Management Server does not start.

If the server starts, and if the preceding options are specified in the Data Management Server's dmserver.cfg file, then the local values supersede the metadata values. For this reason, the preceding options should be commented-out in dmserver.cfg. This happens

by default when you install the DataFlux Data Management Server with the SAS Management Server.

To change the metadata definition of the DataFlux Data Management Server, open SAS Management Console, enter administrative credentials, right-click the Data Management Server instance, and select **Properties**. After you save your changes, restart the DataFlux Data Management Server to download the latest configuration option values.

Configure Server Restart

Because the Data Management Server cannot start unless the SAS Metadata Server is fully operational, you might want to configure a server dependency to prevent failures at invocation. To configure a server dependency, see "Troubleshoot Server Start or Restart".

Additional Configuration after Installation

After you install a DataFlux Data Management Server for use with a SAS Metadata Server, you create new user and group definitions (as needed) on the SAS Metadata Server. To create users and groups on the SAS Metadata Server, see the *SAS Intelligence Platform: Security Administration Guide*.

You can also implement other access controls on the DataFlux Data Management Server. You can restrict server access by IP address, and you can enable ALLOW and DENY groups, as described in "Manage Permissions".

Configure Mid-Tier Options for Security

The Data Management Server uses the following mid-tier configuration options when it connects with a SAS Metadata Server and with the Visual Process Orchestration software. The default values of these options, which are set at install time, are sufficient in most cases. You can set these options in the file ***install-path*/etc/app.cfg**.

Configuration Option	Description
BASE/ APP_CONTAINER_DOMAIN	Specifies the authentication domain that is expected by mid-tier application container services. If this option is not specified, then the value of DefaultAuth is used by default. DefaultAuth specifies a default authentication domain at install time.
BASE/APP_CONTAINER_LOC	Specifies the path to the mid-tier application container services. In most cases this option is not required. If it is required, then the value is typically an HTTP address.

Configuration Option	Description
BASE/AUTH_DEFAULT_DOMAIN	Specifies the name of the resolved identity domain. When a supplied user ID is associated with multiple logins in metadata, the authentication domain for the submitted credentials in the following sequence: 1. Use the value of BASE/AUTH_DEFAULT_DOMAIN, if the value matches the authentication domain of one of the user's logins, or 2. Use the value of DefaultAuth, if the value matches the authentication domain of one of the user's logins, or 3. Use the domain of the first login that matches the submitted login.

Configure a DataFlux Authentication Server for Security

Overview

Use this section to configure your DataFlux Data Management Server to use a DataFlux Authentication Server to support authorization and authentication. To use a SAS Metadata Server for this same purpose, see "Configure a SAS Metadata Server for Security".

To use an Authentication Server for security, you first install, configure, and register the Authentication Server. You then create users and groups on the Authentication Server. When the Authentication Server is configured, you install and configure the DataFlux Data Management Server, develop authorizations, and connect to the Authentication Server.

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

Prepare an Authentication Server

Follow these steps to prepare an Authentication Server to provide authorization and authentication support for a DataFlux Data Management Server:

1. Register your new Data Management Server in DataFlux Data Management Studio, as described in "Register a DataFlux Data Management Server".
2. Install and configure your Authentication Server, and then create domains, users, groups, and shared logins, as described in the *DataFlux Authentication Server Administrator's Guide*.
3. To register your Authentication Server, open DataFlux Data Management Studio and click the **Administration** riser bar.
4. Select **Authentication Server**, and then select **New**.

5. In the Add Authentication Server Definition dialog box, enter server connection information. If you want to connect to this particular Authentication Server by default, each time you start Studio, click **Set as default**.
6. To create a group of administrators for the DataFlux Data Management Server, open the **Group** riser, click **All Groups**, and then click **New Group**.
7. To add one or more users to your new group of administrators, click the new group, and then click **Add Members**.
8. If you need to create a new administrative user definition, click the **Users** riser, and then click **Add New**.

   CAUTION:
   All of the members of your administrative group will be configured to have unrestricted access to all of the objects and commands on this particular DataFlux Data Management Server.
9. Click **Test Connection** and click **OK** twice to close the dialog box.

Configure the DataFlux Data Management Server to Use the Authentication Server

After you prepare Authentication Server, follow these steps to configure the DataFlux Data Management Server.

1. Open the file ***install-path*/etc/dmserver.cfg**.
2. Enable security by setting **dmserver/secure = yes**.
3. Identify the name of your administrative group by setting **dmserver/secure/grp_admin = *your-group-name***.
4. To prevent authorization checks, consider adding users and groups to the options **dmserver/secure/grp_allow** or **dmserver/secure/grp_deny**. For information about these options, see "Manage Permissions".
5. Save and close dmserver.cfg.
6. Specify the Authentication Server by setting **base/auth_server_loc = iom://*server-network-name*:21030**. 21030 is the default port number, as shown in this example:

```
#
# 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://Orion.us.mktng.com:21030
```

7. Save and close dmserver.cfg, and then restart the DataFlux Data Management Server.
8. Open Data Management Studio, click the **Administration** riser bar, and then connect to the DataFlux Data Management Server that you just restarted.
9. In the **Details** pane, notice that the security status has changed to **Secured**. When the server is secured, the **Data Connections** and **Security** tabs are enabled.

Now that security is enabled, you grant permissions on the DataFlux Data Management Server. Note that all users and groups must be created on the Authentication Server

before they can be accessed on the DataFlux Data Management Server. To learn more, see "Manage Permissions".

Manage Permissions

Overview

When security is enabled, each DataFlux Data Management Server maintains a local set of permissions that determine, in part, a user's access to the data sets and commands on that server. Authorization can also be determined by IP address, ALLOW and DENY groups, and access control for the USERS and PUBLIC groups.

Control Access for the USERS and PUBLIC Groups

The USERS and PUBLIC groups are defined by default on the SAS Metadata Server and the Authentication Server. You can set configuration options to enable the USERS and PUBLIC groups to allow or deny access to all DataFlux Data Management Server objects and all object-based commands by default.

Enabling access to USERS or PUBLIC can be helpful when you want to make jobs and services available to a wider audience, without editing default ACLs. When you create a job or service, the default ACL for that object allows only you, the object owner, to run that job or service. If you want to make most of your jobs and services available to a wider audience, then you can enable access for the USERS group. If a smaller percentage of jobs and services need access control, then you can override the USERS grant permission in the ACLs of those jobs and services.

Access control for the USERS and PUBLIC groups is determined by two configuration options. Access is denied to both groups by default. Consider these implementation details before you enable access:

- If you grant access to USERS or PUBLIC, then that access is applied only to object that you create after you set the configuration option. Access by USERS or PUBLIC is not granted to any existing objects. You can grant access to USERS or PUBLIC by updating the ACLs of existing objects.
- Note that these permissions apply only to access rights on a specific object.
- Command permissions that are not object-based, such as List/Post, are not affected by this setting.

Follow these steps to set ALLOW or DENY permissions for the USERS and PUBLIC groups:

1. Open the configuration file ***install-path*/etc/dmserver.cfg**.
2. Set the ALLOW or DENY value for the following configuration options:
   - DMSERVER/SECURE/DEFAULT_ACE_USERS = [ALLOW | DENY]
   - DMSERVER/SECURE/DEFAULT_ACE_PUBLIC = [ALLOW | DENY]
3. Save and close the configuration file, and then restart the DataFlux Data Management Server.

Edit **dmserver.cfg** to set values for these configurations. The default setting for each of these configuration options is **DENY**, which essentially blocks users without explicit

permissions on each object. If set to **ALLOW**, the default ACL created by DataFlux Data Management Server grants access rights to that object for that group set.

Consider the following when using these configuration options:

- The options apply only to PUBLIC and 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 apply only to access rights on a specific object.
- Command permissions that are not object based, such as List/Post, are not affected by this setting.

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

- A user creates an object, a job or a service, on the DataFlux 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 the user the owner of the object. If the user needs to make that object available to other users or groups, the user can explicitly set whatever ACL is needed.
- A user posts numerous jobs or services to the DataFlux 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 Using a Job List

When a user posts a job or service to the server, that user becomes the owner of that object. The owner of an object can always 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.

Follow these steps to grant permissions directly from a job list in DataFlux Data Management Server for Batch Jobs and Real-Time Services:

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

1. Open Data Management Studio and click the **DataFlux Data Management Servers** riser bar.
2. In the left navigation pane, select the DataFlux Data Management Server that you want to work with and connect to that server.
3. Click the + sign next to your server to expand the list of job folders.
4. Click the + to expand the category of jobs or services that you want to work with: **Batch Jobs**, **Real-Time Data**, or **Process Services.**.
5. Select a job or service from list in the left navigation pane, and then click the **Permissions** tab in the right information pane.
6. Under **Participants**, click **Add** to open the Add Users and Groups dialog box.

   Note: If the **Permissions** tab does not appear, you might be viewing a profile job that does not have object-level access control.
7. Select a user, or multiple users, and click **Add**. The user is added to the participant list for the job and granted permissions.

Note: On 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

Follow these steps to remove a user or group object from the **Users and Groups** list on the DataFlux Data Management Server:

Note: The definition of the user or group is retained on the SAS Metadata Server or the Authentication Server.

1. Connect to 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 box.

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.*

Permission	Description
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 service	When enabled, the user can list real-time process services.
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

Specify the following configuration options to control access to the DataFlux Data Management Server by IP address. The options are specified in ***install-path*/etc/dmserver.cfg**.

Option Syntax	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 DataFlux 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 DataFlux 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 hyphen character ('-') between the low and high ends of the range.

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

Configure SSL and AES

Overview

Beginning in the 2.5 release, the DataFlux Secure software is installed by default when you install your DataFlux Data Management Server. The DataFlux Secure software provides increased security through the Advanced Encryption Standard and through the use of the Secure Sockets Layer to protect HTTP client connections. These security enhancements, and their configuration on the DataFlux Data Management Server, are addressed in detail in the *DataFlux Secure Administrator's Guide*.

All of the clients and servers that connect to the DataFlux Data Management Server need to be configured for the same level of encryption and SSL implementation.

Enable SOAP with SSL

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

CAUTION:

Stop the DataFlux Data Management Server before you make any changes to the configuration file.

Configuration Option	Description
DMSERVER/SOAP/SSL	If you use a DataFlux Authentication Server for security, then set the value to YES. Later, if you need to disable SSL, set the value to NO. If you use a SAS Metadata Server for security, then this option should remain disabled by comment characters, as is the case by default. This option should not be set in dmserver.cfg because the value is set at server start, based on the server's metadata definition. If you set the option locally, then the local value overrides the value in metadata.
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, then comment-out this option. The value of this option must be encrypted. To encrypt passwords, see "Encrypt Passwords for DSNs and SSL".
DMSERVER/SOAP/SSL/ CA_CERT_FILE	Specifies the file that stores your trusted certificates.
DMSERVER/SOAP/SSL/ CA_CERT_PATH	Specifies the path to the directory where you store your trusted certificates.

Configure OpenSSL

In the UNIX and Linux operating environments, you are required to install OpenSSL if you have not done so already. If you install OpenSSL, be sure to check that all of the DLLs are installed in the correct locations.

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. If the DLLs are *not* copied to System32, choose one of the following steps:

1. Manually copy the DLLs to System32.
2. Copy the DLLs to the ***install-path*\bin** directory of DataFlux 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.

Encrypt Passwords for DSNs and SSL

Overview

To improve security, encrypt the passwords of your DSNs and your SSL key file.

Encrypt in Windows

To encrypt passwords in the Windows operating environment, run ***install-path*\bin\EncryptPassword.exe**. Enter the password, confirm your initial entry, and receive the encrypted password.

Encrypt in UNIX and Linux

To encrypt passwords in the UNIX and Linux operating environments, enter the command **dmsadmin crypt**.

Troubleshoot Security Errors

Overview

Interpret and resolve the following security errors.

401 Unauthorized

This HTTP error can indicate that the user entered incorrect credentials. The error can also indicate that a user account has not been created on the authorizing server (SAS Metadata Server or DataFlux Authentication Server.)

403 Forbidden

This HTTP error indicates that the user is not authorized to use a particular Data Management Server command. For more information, see “Manage Permissions”.

Chapter 4

Administering the DataFlux Data Management Server

Start or Stop the Runtime Server on Windows

In the Windows operating environment, to start and stop the SAS Visual Process Orchestration Runtime Server, use the Microsoft Management Console or follow these steps:

1. Select ⇨ **Start** > **Control Panel**.
2. Double-click **Administrative Tools** ⇨ **Computer Management**.
3. Expand the **Services and Applications** folder.
4. Click **Services**.
5. Click **SAS Visual Process Orchestration Server (Runtime)**.
6. Click either **Stop the service** or **Restart the service**.

Note: You can also access the service by selecting **Start** ⇨ **All Programs** ⇨ **DataFlux** .

If the Data Management Server fails to start or restart, see "Troubleshoot Server Start or Restart" on page 34.

Start or Stop the Runtime Server on UNIX or Linux

In the UNIX or Linux operating environments, use the following command to stop or start the DataFlux Data Management Server:

```
install-path/bin/dmsadmin your-command
```

Here is a typical example:

```
<SASHome>/DataFluxDataManagementServer/2.5/dmserver03/bin/dmsadmin start
```

The dmsadmin utility accepts the following options:

Command	Description
start	Starts the server. For example: `./bin/dmsadmin start`
stop	Stops the server.
status	Checks whether the server is running.
help	Displays Help information.
version	Displays version information.

If the Runtime Server fails to start or restart, see “Troubleshoot Server Start or Restart” on page 34.

Troubleshoot Server Start or Restart

If your Data Management Server fails to start or restart, you might need to resolve a server dependency. This dependency applies when the DataFlux Data Management Server is configured to use a SAS Metadata Server for authorization and authentication. The SAS Metadata Server needs to be fully operational before the Data Management Server can start. This server dependency exists because the DataFlux Data Management Server needs to retrieve several configuration option values from the SAS Metadata Server at start-up.

The server dependency occurs predominantly in single-machine installs, when all services start at one time.

You can resolve the server dependency as you see fit, or you can run the following command on the host of the DataFlux Data Management Server:

```
sc config "DMServer-service-name"
      depends= "SASMetadata-service-name"
```

The service names are specified in the properties of the service. Do not use the displayed server names.

Use quotation marks as shown, use no blank space after **depends**, and use a blank space after **=**, as shown in the following example:

```
sc config "dfx-DMServer-server1"
      depends= "SAS [Config-Lev1] SASMeta - Metadata Server"
```

Administer DataFlux Data Management Server Log Files

All of the service requests that are received by the DataFlux Data Management Server are assigned a unique request identification (RID). As the DataFlux Data Management Server processes a request, all log entries for that request begin with the associated RID. Log events that relate to security use a different format.

By default, a new server log subdirectory is generated for each server request. The default path to the log files is:

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

The default subdirectory name is defined as shown in this example:

```
20110804-14.26-pid5072__034C24
```

20110804 is the date, **14.26** is the time, **pid5072** is the process ID, and **034C24** is a unique Data Management Server request ID.

Use the following configuration options in dmserver.cfg to change the default logging behavior:

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 is shown above.
DMSERVER/NO_WORK_SUBDIRS = Yes | No	Controls whether each server run creates a new log subdirectory. The default is **No**, which specifies that all log and work files are created in subdirectories. To disable creation of the subdirectories change this value to **Yes**.

To change the storage location or logging level for dmserver.log, open the file ***install-path*\etc\dmserver.log.xml**. To change the location of the log, change the option BASE/LOGCONFIG_PATH. To change the logging level, see "Change Log Events and Thresholds".

To change the encoding of your server job, set the configuration option BASE/JOB_LOG_ENCODING in the file ***install-path*/etc/app.cfg**. By default, the log is written in the encoding of the locale of the process that executes the job. For English-speaking organizations, the encoding can be LATIN-1 or UTF-8. If a log line contains characters that cannot be represented in the encoding, then the log line is not written to the log file.

Administer Data Service Log Files

When enabled, a unique data service log file records log events for each server request that runs a real-time data service (as processed by DFWSVC.) The name of each data service log file is added to the DataFlux Data Management Server log file dmserver.log. The server log also contains debugging information for the job run of the real-time data service. If a new service process is used, then the server log includes the PID of the corresponding process.

Note: To maximize performance, data service logging is disabled by default. To enable data service logging for testing purposes, follow the steps provided later in this section.

Each new data service log file is stored by default in the following directory:

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

The name of the log subdirectory specifies the date, time, process ID, and server request ID.

The name of the data service log file is illustrated in the following example:

```
10.52.24.226_2778_datasvc_Verify-Address-Job.ddf.log
```

In the preceding example, **10.52.24.226** is a time stamp, **2778** is the server request ID, and **datasvc** is the log file type. The remainder of the name specifies the name of the real-time service.

Data service logging is configured by default by the following file:

install-path/etc/service.log.xml

Follow these steps to enable logging for real-time services:

1. Open service.log.xml and locate the **root** tag.
2. Change the **OFF** value in <level value="OFF"/> to **DEBUG** or **TRACE**, depending on the level of information that you want to gather. TRACE provides the greatest level of information.
3. Restart the Data Management Server.
4. At the conclusion of testing, repeat this procedure to disable logging.

If you require additional information to conclude your testing process, contact your SAS technical support representative.

To change the name and location of the data service log configuration file service.log.xml, open the following file:

install-path/etc/service.cfg

In service.cfg, change value of the option BASE/LOGCONFIG_PATH, and then restart the Data Management Server.

To change the encoding of the job logs for your real-time data services, set the configuration option BASE/JOB_LOG_ENCODING in the file ***install-path*/etc/app.cfg**. By default, the log is written in the encoding of the locale of the process that executes the job. For English-speaking organizations, the encoding can be LATIN-1 or UTF-8. If a log line contains characters that cannot be represented in the encoding, then the log line is not written to the log file.

Administer Log Files for Batch and Profile Jobs

A unique log file is generated by default for each batch and profile job. Each such job is processed by a unique instance of the DFWFPROC process. The name of each batch job log file is added as an entry in the Data Management Server log file dmserver.log. Job status, based on the content of the job logs, is displayed in the Monitor folder in DataFlux Data Management Studio.

Note: You can use the SAS Job Monitor in the SAS Environment Manager to collect statistics during and after job runs as described in "Collect Job Status Information with the SAS Job Monitor".

Each new job log file for a batch or profile job is stored by default in the following directory:

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

The default directory is specified by the configuration option DMSERVER/WORK_ROOT_PATH.

The name of the log subdirectory specifies the date, time, process ID, and server request ID.

The name of the data service log file is illustrated in the following example:

```
18.00.24.125_3727_wfjob_Marketing_Profile_1.log
```

In the preceding example, **18.00.24.125** is a time stamp, **3727** is the server request ID, and **wfjob** is the log file type. The remainder of the name specifies the name of the job.

Batch and profile job logging is configured by default by the following file:

install-path/etc/batch.log.xml

To change the default name and location of the batch job log configuration file, edit the value of the option BASE/LOGCONFIG_PATH. To change log events and thresholds, see "Change Log Events and Thresholds". Restart the server to apply your changes.

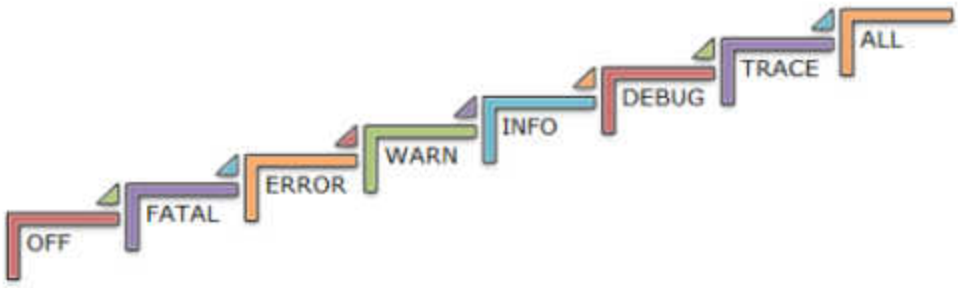
To change the encoding of your batch and profile logs, set the configuration option BASE/JOB_LOG_ENCODING in the file ***install-path*\etc\app.cfg**. By default, the log is written in the encoding of the locale of the process that executes the job. For English-speaking organizations, the encoding can be LATIN-1 or UTF-8. If a log line contains characters that cannot be represented in the encoding, then the log line is not written to the log file.

Change Log Events and Thresholds

Loggers and appenders determine the content in DataFlux Data Management Server log files. The loggers and appenders are defined in each log configuration file, such as ***install-path*\etc\dmserver.log.xml**.

Appenders specify the log output destination. Loggers specify log event types and thresholds. If a logger lists a given log event, then those events are recorded in the log file. The threshold value determines the amount of information that is captured in the log

file for each event. The available threshold levels are ranked as shown in the following diagram.



The default threshold level capture 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, contact your SAS technical support representative for assistance.

Altering threshold levels above INFO when the server is operational in a production environment is discouraged since this can result in a reduction in server performance.

When you change a log configuration file, you are required to restart the Data Management Server.

To learn more about logging, see the *SAS Logging: Configuration and Programming Reference* and the *SAS Interface to Application Response Measurement (ARM): Reference*.

Troubleshoot Server Start

If the Data Management Server does not start, and if the server log file lists the failure dfwlpListenAttr_connattr(wlp), then a port might be in use by another application. By default, SOAP requests are handled on port 21036. WLP requests are handled on port 21037. If either of the default ports are being used by another application, then assign that process to an unused port.

If the server log does not indicate the source of the problem, then follow these steps if your server installed on Windows:

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**. Typically, two such events are logged for each time period. One message specifies the name of the log file, as shown in the following example:

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

If your server is installed on UNIX or Linux, then errors will be written to the stdout location of the shell from which the DataFlux Data Management Server was started.

Troubleshoot ActiveX Error to Display Help

Internet Explorer versions 6 and later can be configured to block the download and execution of ActiveX controls. ActiveX is required to display the online Help for DataFlux Data Management Studio. Follow these steps to enable the download and execution of ActiveX controls:

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

Chapter 5

Managing Data Connections

Overview of Data Connections

The Data Management Server connects to data sources on databases through ODBC or through jobs that connect to the Federation Server. For more information about accessing data on or through a Federation Server, refer to the *DataFlux DataFlux Data Management Studio User's Guide* and to the *DataFlux Federation Server Administrator's Guide*.

To add a data source using ODBC, use the ODBC Data Source Administrator provided with Windows, or use the dfdbconf command in UNIX and Linux.

You can configure the following data connections (also known as DSNs or data sources) for the jobs that you run on the DataFlux Data Management Server:

- **Domain-Enabled ODBC Connection** - Enables you to create a connection that links a DataFlux Authentication Server domain to an ODBC Data Source Name (DSN). User credentials from the Authentication Server are automatically applied

when the user accesses the domain-enabled connection. This approach ensures that the appropriate credentials for that domain are applied to the access request.

- **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 SAS 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.

In Windows, DSNs are stored in ***install-path*\etc\dftkdsn**.

In UNIX and Linux, DSNs are stored in ***install-path*/etc/odbc.ini**.

You can store ODBC credentials for data sources that require login credentials with the **ODBC Credential Manager**. With stored ODBC credentials, you can make connections to data sources without being prompted for login credentials. When a job is run, the saved user credentials are retrieved and used. The credentials are not stored within the job. The job references the connection by DSN only. In UNIX and Linux, credentials are stored in the directory **/$HOME/.dfpower/dsn**.

When you develop jobs and services in DataFlux Data Management Studio, use the Data Connections riser to set up and store login credentials for any Open Database Connectivity (ODBC) data source. The DataFlux Data Management Server can use these data sources directly if Studio is installed on the same host as DataFlux Data Management Server.

Stored credentials do not have to be entered each time the job is run, and that information can be used by any DataFlux application. If you do not use stored credentials, then your job must authenticate through a Metadata Server or Authentication 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.

If you want to use ODB drivers other than those that are supplied, note that the Data Management Server is compatible with most ODBC-compliant data sources. Also note that SAS provides limited support for drivers that are not supplied by SAS.

If you develop jobs that access a SAS Federation Server, then you can use JDBC drivers and other drivers that are written for native access to popular databases. To learn more about developing jobs that use Federation Server drivers, refer to the *DataFlux Data Management Studio User's Guide* and to the *SAS Federation Server Administrator's Guide*.

Configure the Data Access Component (DAC)

The Data Access Component (DAC) allows the DataFlux 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. In the Windows operating environment, ODBC DSNs are managed by the Microsoft ODBC Administrator. In the UNIX and Linux operating environments, ODBC DSNs are created with the dfdbconf tool and tested with the dbdfview tool. TKTS DSNs are managed by the DAC. TKTS DSNs stored in the DSN directory.

The DAC is configured with the following two options in the DataFlux Data Management Server's app.cfg file. If necessary, the options can be moved to the macro.cfg file.

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

For more information about 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 DataFlux Data Management Studio Online Help*.

Display Data Connections

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

1. Open Data Management Studio and click the **Administration** riser.
2. Connect to the Data Management Server.
3. Click the **Data Connections** tab to display data connections.

Create a Server Job That Uses a Driver and a Data Source

Follow these steps to develop a job that runs on a DataFlux Data Management Server and accesses a database:

1. Install the DataFlux ODBC drivers (unless they were installed initially) on both the Data Management Studio client and on the DataFlux Data Management Server. Use the regular installers as needed. DataFlux provides a number of wire-protocol drivers that enable you to connect to specific databases without using a client-side library.
2. Create a DSN on the database server.
3. Configure an ODBC client-side connection on the DataFlux Data Management Studio host using the **ODBC Connections** window in Studio, as described in the *Data Management Studio User's Guide*.
4. Create and test the job in DataFlux Data Management Studio. The nodes in the job will access the ODBC connection.
5. Upload the job from Studio to the Data Management Server.
6. Either copy the ODBC connection file onto the server or create the connection on the server. To create a server connection in Windows, use the **ODBC Data Source Administrator**. If your server is running in UNIX or Linux, use the **dbdfconf** tool that is provided with the DataFlux Data Management Server.

If DataFlux Data Management Studio and DataFlux Data Management Server are installed and running on the same (Windows) host, then you will need to set up the ODBC DSN two times. Set up one DSN through **ODBC Connections** in Studio. For the DataFlux Data Management Server, set up the DSN again using the **ODBC Data Source Administrator**.

Use the Windows ODBC Data Source Administrator

In the Windows operating environment, use the Microsoft ODBC Data Source Administrator to manage database drivers and data sources on a DataFlux Data Management Server.

Note: To learn about DSN options, open the Help for the ODBC Data Source Administrator.

Follow these steps to configure a driver to connect to a database:

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 box and click **OK** when finished.

Use dfdbconf and dfdbview for UNIX and Linux ODBC Connections

Follow these steps to use the UNIX and Linux ODBC configuration tools dfdbconf and dfdbview:

1. Execute the following command:

   `install-path/bin/dfdbconf`

2. Select a driver from the list of available drivers.
3. Set the appropriate parameters for the driver. The new data source is added to the odbc.ini file.

   Note: You can also use dfdbconf to delete the data sources.

4. Execute the following command to test your new data source:

   `install-path/bin/dfdbview data-source-name`

   You will be prompted for a user ID and password if the connection is secured.

5. If the connection succeeds, use the prompt to enter SQL commands to test the connection. If the connection fails, resolve the errors described in the error messages.

Create a Domain-Enabled ODBC Connection

Domain-enabled ODBC connections use a domain and credentials that are supplied by a SAS Metadata Server or a DataFlux Authentication Server. You can create domain-enabled ODBC connections for use with jobs that run on the Data Management Studio host, or that run on a DataFlux Data Management Server host.

If you create a domain-enabled ODBC connection for use with Studio, and if you want to move that connection to a DataFlux Data Management Server, then you copy the .**dtfk** file from the client to the DataFlux Data Management Server into the directory ***install-path*\etc\dftkdsn**.

To execute jobs, the DSN in the domain-enabled ODBC connection must be defined in a location that is accessible to the DataFlux Data Management Server. Otherwise, the reference to the DSN in the domain-enabled ODBC connection will not resolve.

To create a new domain-enabled ODBC connection for use on the DataFlux Data Management Studio host, see the *Data Management Studio User's Guide*.

To create a new domain-enabled ODBC connection for use on the DataFlux Data Management Server host, follow these steps:

1. Open a registered Data Management Server using the riser bar in Studio and log on if prompted.
2. Click the **Data Connections** tab and click **New**.
3. Select **Domain Enabled ODBC Connection** from the drop-down menu.
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 and connection in the **Description** field.
6. Select the DSN.
7. In the **Domain name** field, enter the actual domain name as it was created in the Metadata Server or Authentication Server.
8. Click **OK**.

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

Create a Custom Data Connection

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

Follow these steps to create a custom connection:

1. In DataFlux Data Management Studio, click the **Data** riser, and then expand **Data Connections** .
2. Click **New Data Connection**, and then select **Custom Connection**.
3. Enter the name, description, and the full connection string.

4. Click **Test Connection**.
5. 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. The **Default** value assigns read-write access. The **Read-Only** value assigns Read-Only access. The **Temp** value specifies that the data set is to be treated as a scratch file, which is stored only in memory, not on disk.
6. Verify that the appropriate option is specified in the **Compression** field. If you compress data, you might experience a slowdown in performance. The **No** value specifies that the observations are uncompressed. Use the **Yes** or **Character** value to compress character data. Use the **Binary** value to compress binary data.
7. Verify that the appropriate option is specified in the **Table Locking** field. The **Share** value specifies that other users or processes can read data from the table but prevents other users from updating. The **Exclusive** value locks tables exclusively, which prevents other users from accessing that table until you close it.
8. Verify that the appropriate option is specified in the **Encoding** field. The default value is 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 enter the first few letters of the desired encoding, the closest match will be added to this field.
9. The **Connection String** field displays the connection string for the SAS data set. Check the connection string to see whether 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. In DataFlux Data Management Studio, click the **DataFlux Data Management Servers** riser to access a list of DataFlux Data Management Servers.
2. Select the name of the server for which you want to manage connections. If you are prompted to do so, enter 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 box that opens, make the desired changes and click **OK**.

Delete a Data Connection

Follow these steps to delete a data connection:

1. From Data Management Studio, click the **DataFlux Data Management Servers** riser to access a list of your DataFlux Data Management Servers.
2. Click the name of the server for which you want to manage connections. If you are prompted to do so, enter 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**.

Manage ODBC Credentials

ODBC credentials enable connections to data source. The credentials contain connection information, so they do not need to be stored inside the job, or entered when job is run. This allows for better protection and management of security, increased confidentiality, and a more versatile way to handle access to data sources that require authentication.

To manage ODBC credentials, complete the following steps:

1. In DataFlux Data Management Studio, click the **DataFlux Data Management Servers** riser bar.
2. Select the name of the DataFlux Data Management Server for which you want to manage connections. If you are prompted to do so, enter your user ID and password, and then click **Log On**.
3. In the information pane, select the **Data Connections** tab and click the **Manage ODBC Credentials** icon.
4. To create ODBC credentials in the Manage ODBC Credentials dialog box, click **New ODBC Credentials**. Enter the ODBC DSN, user name, and password. Review your entries, and then click **OK**.
5. To edit ODBC credentials, select a name from the list and click the **Edit ODBC Credentials** icon. In the **ODBC Credentials** dialog box, change the user name or password that will be used to access the ODBC DSN. Click **OK** to close the dialog box. Note that the **Edit ODBC Credentials** icon is available only when credentials have been saved for an ODBC DSN.
6. To delete ODBC credentials, select a name and click **Delete ODBC Credentials**. You can use **Ctrl** + **left click** to select more than one name. Click **OK** to close the

Manage ODBC Credentials dialog box when you are finished. Use caution when deleting an ODBC credential. When a name is deleted from the list, clicking **Cancel** will not reverse the deletion.

Troubleshoot ODBC Data Connections

Overview

If your ODBC connections show any of the following symptoms, refer to the following resolutions.

SQL Server ODBC Driver Error in Windows

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

Note: Use the 32 or 64-bit driver depending on the operating environment of the DataFlux Data Management Server. Access the drivers by selecting **Control Panel** ⇨ **ODBC Data Sources**. The DataFlux drivers are listed in the **Drivers** tab.

Teradata and Informix ODBC Drivers Fail to Load

In the Solaris x86 operating environment, DataDirect does not currently provide Teradata or Informix drivers.

In Linux operating environments, the directory that contains the Teradata client libraries needs to be in your LD_LIBRARY PATH. The exact path will vary depending on the version of your Teradata client.

Chapter 6

Managing Jobs, Services, and the Repository

Overview of Jobs and Services

The types of jobs and services that you can store and run on a DataFlux Data Management Server include real-time data services, real-time process services, batch jobs, and profile jobs.

Real-Time Services Data Services

Real-time data services are designed to quickly respond to a request from a client application. Real-time data services can process a small amount of data input from a client, or it can retrieve and deliver a small amount of data from a database. Real-time data services are executed by the DFWSFC process, as defined in "Manage the DFWSVC Process".

Real-Time Process Services

Real-time process services accept input parameters only from clients, to trigger events or change a display. Real-time process services are executed the DFWFPROC process, which runs a WorkFlow Engine (WFE), as defined in "Manage the DFWFPROC Process".

About Real-Time Data Services and Real-Time Process Services

If a real-time data service or real-time process service fails to terminate normally, then the service is terminated when the client connection times-out.

To maximize performance, logging is not enabled for real-time services. To activate logging for debugging purposes, see "Administer Data Service Log Files".

Real-time services are stored in ***install-path*\ var\data_services | process_services**.

Batch Jobs
: Batch jobs are designed to be run at specified times to collect data and generate reports. Batch jobs are not intended to provide real-time responses to client requests.

  All batch jobs are logged in dmserver.log. For more information, see "Administer Log Files for Batch and Profile Jobs".

  Batch jobs are stored in ***install-path*\ var\batch_jobs**.

  Batch jobs, like real-time process services, are run by the DFWFPROC process. You can pass input parameters into batch jobs, but not any actual data.

Profile Jobs
: Profile jobs are designed to analyze the quality of specified data sets. Profile jobs are handled as repository objects. They are required to reside in the Data Management Repository. When you run a profile job, the server finds the job in the repository and then starts a new instance of the DFWFPROC process. The requested profile is then run by ProfileExec.djf, which resides in the same directory as the repository. For more information about the Data Management Repository, see "Create a Repository".

  Unlike batch jobs, you cannot grant unique user permissions for profile jobs since they do not have associated object-level access control. To learn more about permissions, see "Manage Permissions".

  When you install a new version of the DataFlux Data Management Server, you are required to import all of your profile jobs into a new Data Management Repository. For more information about importing profile jobs, see "Migrate Security after Software Upgrades".

Note that in Windows, it is possible to save a DataFlux Data Management Server job to a directory, and then not be able to see that job in that directory. To resolve this issue, save your jobs 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 maps those drives.

Job names in Data Management Server can use alpha numeric characters and the following characters:

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

The maximum length of job names is 8,192 bytes.

In UNIX or Linux, to run a shell command in a job, use the execute() function, as shown in the following examples. To run the command directly:

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

To run a command through a shell:

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

The preceding examples return the host authorizations for a text file.

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.

The SOAP server uses a SOAP interface, as defined in the DataFlux Web Service Definition Language (WSDL) file. For more information about the WSDL file, see "Customize the Server's WSDL File".

The 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 by omitting Connection: close (for HTTP 1.1). The connection parameters are specified in the HTTP header of the SOAP request.

To manage the configuration of the SOAP and the WLP servers, set the following configuration options.

Configuration Option	Description
DMSERVER/SOAP/LISTEN_PORT	Specifies the port on which the SOAP server listens for connections. The default value is 21036 when you use a DataFlux Authentication Server for security. When you use a SAS Metadata Server for security, the DataFlux Data Management Server uses the DMSERVER/NAME option to retrieve from metadata the values of three configuration options, including LISTEN_PORT. If the SAS Metadata Server does not return a value for LISTEN_PORT, then the DataFlux Data Management Server does not start. If a value is returned, and if dmserver.cfg also contains a value for LISTEN_PORT, then the local value overrides the metadata value. For this reason, it is recommended that you not set LISTEN_PORT in dmserver.cfg when using a SAS Metadata Server. For further information, see DMSERVER/NAME and DMSERVER/SOAP/SSL.
DMSERVER/WLP	Enables or disables the WLP server. When the value is YES, the WLP server is started and uses its own listen port. When the value is NO, the WLP server is bypassed during start-up of Data Management Server. This means that WLP clients cannot connect to the DataFlux Data Management Server, but SOAP clients can. The DataFlux Data Management Server log receives entries for the status of WLP server.

Configuration Option	Description
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 host name or IP address to which the WLP server must bind. By default, this option is left blank. For more information, see DMSERVER/SOAP/LISTEN_HOST.

Configure the Server to Pre-load Services

Overview

The following sections describe how to use pre-load configuration settings when you start your DataFlux Data Management Server. This is helpful if you typically use the same services each time you run DataFlux Data Management Server.

Use the following options to configure pre-load:

- DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL = *count*
- DMSERVER/SOAP/DATA_SVC/PRELOAD = *count*:*name-of-service*count:*name-of-service* ...

  The value *count* specifies the number of pre-load instances. The value *name-of-service* indicates the name of the service element. This can include the directory where the service is located.
- DMSERVER/SOAP/DATA_SVC/PRELOAD_DURING_RUN = *yes*|*no*

  By default, the Data Management Server pre-loads all configured services before accepting SOAP requests. When the value is *yes*, the DataFlux Data Management Server starts a separate thread to pre-load all configured services at run time, while accepting SOAP requests at the same time. If DataFlux Data Management Server is stopped while the pre-load thread is still running, that thread will be terminated.

Pre-load All Services

The configuration option DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL = *count* causes the DataFlux Data Management Server to find and pre-load a specified number of all services. 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 DataFlux Data Management Server to preload two instances of each service that is available, including those found in subdirectories.

Pre-load One or More Specific Services

The configuration option DMSERVER/SOAP/DATA_SVC/PRELOAD =*count:name-of-service* designates the specific services, as well as the count for each service, that the DataFlux Data Management Server is to pre-load at start-up. Use additional count and service elements for each service. Separate each count and service element by one or more blank spaces. The service element itself cannot include blank spaces. Also, all elements must be listed on a single line. Using this format, you can configure a directive that starts a number of services, each with a different count.

The following example loads two counts of the abc service and one count of xyz service. The xyz service is located in the subdir2 subdirectory:

```
DMSERVER/SOAP/DATA_SVC/PRELOAD = 2:abc.ddf 1:subdir1\xyz.ddf
```

Configure Complex Pre-loads

By combining options, you can configure more complex pre-loads. The two options add the counts arithmetically to determine how many services are actually loaded. Internally, the DataFlux Data Management Server builds a list of all of the services that it needs to pre-load 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 option instructs the DataFlux Data Management Server to pre-load a total of two instances of all existing services. The second options modify the first as follows:

- 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 DataFlux Data Management Server tries to preload.
- The svc2.ddf file, which is found in the 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 *count* value must be greater than zero for a service to be pre-loaded.

Some important points to remember:

- DataFlux Data Management Server attempts to pre-load a single instance of all requested services before trying to pre-load 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.
- The *count* value can be a negative value. This is meaningful only when both configuration options are used together.
- Pre-loading stops when the Data Management Server has attempted to pre-load 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 DataFlux 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 DataFlux 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

Overview

The Data Management Server supports a number of SOAP commands that enable clients to run jobs and services and administer the server. These Simple Object Access Protocol (SOAP) commands cause the server to return 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, in ***`install-path`*`/share`**.

Note: WSDL 2.0 is not supported.

SOAP Commands Reference

The following table describes the SOAP commands that appear in the WSDL file:

Command	Description	Command Type
GenerateWSDL	Generates a WSDL for a real-time data service or a real-time process service, or for an entire directory including all associated subdirectories. You can pass service names or a single directory name but both cannot be used together in the command. Using a service and directory name together will result in an error. The DataFlux Data Management Server can generate multiple WSDLs within a request and will not stop if an error occurs. To review the responses that result from the use of this command, see "Response from the GenerateWSDL Command".	
GetServerVersion	Returns the server version and the versions of the installed reference data, the repository, and some of the libraries. Also returns date and time elements.	Single version command
ArchitectServiceParam	Returns the input and output fields of a real-time data service.	Data services command

Command	Description	Command Type
ArchitectService	Runs a real-time data service. The *timeout* integer element specifies the number of seconds to allow a real-time data service to run before it is stopped by the server. If the *timeout* element is omitted from the request, or if the value is 0, then the real-time data service will not be stopped by server. *Note:* The actual duration of a real-time data service can vary depending on the rounding-up of fractional *timeout* values. For example, a *timeout* value of 1.5 is rounded up to an actual duration of 2 seconds.	Data services command
ArchitectServicePreload	Starts the requested number of processes, and loads into those processes the specified real-time data services. For information about preloading services, see "Configure the Server to Pre-load Services".	Data services command
ArchitectServiceUnload	Terminates the process that is running the specified real-time data or process service. For further information,	Data services command
LoadedObjectList	Returns a list of running real-time data service processes, along with the name of the loaded service job for each process.	Data services command
MaxNumJobs	Sets the maximum number of concurrent processes that can run real-time data services. This is a run time setting only and has no effect on the value of the configuration option in the dmserver.cfg file. To learn about configuration options, see "Configuration Options Reference for dmserver.cfg".	Data services command
WorkFlowJobParams	Returns the inputs and outputs of either a rea-time process service or a batch job.	Process services command
WorkFlowService	Runs a real-time process service.	Process services command
UnloadProcesses	Kills all idle dfwfproc processes and subsequent busy dfwfproc processes once they become idle. The DFWFPROC processes run real-time process services and batch jobs.	Process services command

Command	Description	Command Type
RunArchitectJob	Runs a batch job.	Batch and profile jobs commands
RunProfileJob	Runs a profile job.	Batch and profile jobs commands
TerminateJob	Terminates a running batch job or profile job. The client can still retrieve the status, log, and statistics file (if one exists) after the job has been terminated.	Batch and profile jobs commands
JobStatus	Returns status information for one or more batch jobs or profile jobs. Applies to jobs that are running or that have already finished.	Batch and profile jobs commands
JobNodesStatus	Returns status information for every node in a batch job. Applies only to the jobs that are currently running.	Batch and profile jobs commands
JobLog	Returns the log file and statistics file (if one exists) for a batch job or profile job. Applies only to already finished jobs.	Batch and profile jobs commands
DeleteJobLog	Deletes the job log, statistics file (if one exists), and all history for a given job run. Applies only to already finished jobs.	Batch and profile jobs commands
ObjectList	Retrieves a list of available objects of a specified type (data services, process services, batch jobs, or profile jobs).	Object files commands
PostObject	Uploads 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	Downloads an object of a specified type.	Object files commands
DeleteObject	Deletes an existing object of a specified type.	Object files commands
ListAccounts	Returns a list of user and group IDs with explicitly configured server commands permissions (which are included).	Security commands
SetPermissions	Sets server command permissions for a user or group ID.	Security commands

Command	Description	Command Type
DeleteAccount	Deletes server command permissions for a user or group ID.	Security commands
SetACL	Sets an access control list (ACL) for an object (data service, process service, or batch job).	Security commands
GetACL	Retrieves an ACL for an object.	Security commands

Response from the GenerateWSDL Command

When a client requests the GenerateWSDL command, the response contains two lists:

1. A list of the job names for which WSDLs were generated successfully.
2. A list of job names for which WSDLs could not be generated. Each entry in this list includes a detailed error message as to the cause of the problem.

Generating a WSDL is similar to obtaining properties for a service, so the error messages in the job list are similar. Typical messages include the following:

- job file not found
- access denied
- job format is invalid
- failed to connect to DB

The two job lists are sent only when the request is completed, with or without errors. The amount of time that is required to generate a WSDL is determined by the nature of the job and its dependencies.

Reference for WSDL Configuration Options

The following table describes the configuration options in dmserver.cfg that relate to WSDLs.

Configuration Option	Description
DMSERVER/SOAP/WSDL	Specify a value of YES to load existing WSDLs when you start the DataFlux Data Management Server. Specifying YES also enables the server to recognize other WSDL configuration options and respond to client requests for WSDL-based jobs and services. The default value NO specifies that no existing WSDLs are loaded at start-up, no new WSDLs are generated, client requests to run WSDL-based jobs and services are ignored, and other WSDL configuration options are ignored.

Configuration Option	Description
DMSERVER/SOAP/WSDL/GEN	Allows or denies the server the ability to generate run-time WSDLs. Valid option values are NO, SINGLE, or MULTIPLE. The default value NO specifies that the only WSDLs that can be used to respond to client requests are those that are loaded when you start the Data Management Server. Errors are generated in response to requests to the GenerateWSDL command or the PostObject command. The value SINGLE enables the generation of a single WSDL for each instance of GenerateWSDL or PostObject. The value MULTIPLE enables the generation of multiple WSDLs, for SOAP commands that apply to multiple job files or directories of jobs. Note that WSDL generation can be a time-consuming and resource-intensive process. Also note that erroneous requests to generate multiple WSDLs can cause a severe degradation in server performance.
DMSERVER/SOAP/WSDL/GEN_ON_GET	Allows or denies WSDL generation as part of an HTTP getWSDL request. The default value NO indicates that an error is returned if the WSDL does not exist, or if the mod.time stamp in the WSDL differs from the mod.time stamp in the service job file. The different mod.time values indicate that the WSDL has been updated on the server, but not on the client. The value YES indicates that the server does not return an error message if the WSDL does not exist or if the mod.time stamps differ. Instead, the server attempts to generate the latest WSDL. If successful, the server returns the new WSDL to the client. This option is valid only when the value of DMSERVER/SOAP/WSDL/GEN is SINGLE or MULTIPLE. If the value of DMSERVER/SOAP/WSDL/GEN is NO, then the GEN_ON_GET option is ignored and WSDLs are not generated on HTTP getWSDL requests.

Configuration Option	Description
DMSERVER/SOAP/WSDL/RUN_IGNORE_MTIME	When the server receives an HTTP RunSVC request, this option determines whether the client needs to be rebuilt in order to match an updated WSDL. A rebuild is indicated when the mod.time stamp in the WSDL differs from the mod.time stamp in the service job file. The default value NO specifies not to ignore differences in mod.time values. When the server detects a difference in mod.time values, the server returns to the client the error message Job Has Changed. The client responds by requesting that the server regenerate the WSDL and send the WSDL to the client. The client then rebuilds as directed by the new WSDL. A value of YES indicates that the server will not compare mod.time values. The server passes the RunSVC request directly to the service process for execution. This behavior applies to service requests that are based on generic WSDL

Debug Real-Time Services Using SOAP Fault Elements and Log Files

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

The Data Management Server generates the following fault codes:

Error	Description
VersionMismatch	Pertains to the SOAP Envelope element, which is the root element for all SOAP messages. Indicates that the recipient of a message did not recognize the namespace name of the Envelope element.
MustUnderstand	Indicates that the recipient of an element child of the Header element had a **`soap:mustUnderstand`** attribute that was not understood by the recipient.
SOAP-ENV:Client	Indicates that 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 a server problem. The message contents are not at fault. Instead, a resource was unavailable or process logic failed.

Other fault elements are delivered in the SOAP response, such as the tags faultactor or detail. These elements do not specify a reason for the fault or indicate the data element that triggered the error. These tags are generic in nature and are usually returned for any and all SOAP requests. Because the DataFlux 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. To learn more about logs, see "Administer DataFlux Data Management Server Log Files" and see "Administer Data Service Log Files"..

Also note that the nil tags are unused for SOAP fault messaging. It is best not to refer to these elements for problem determination.

Define Macros

Overview

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

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

On a UNIX system:

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

The example macro is useful when you are porting jobs from one machine to another, because the paths to an input file in different operating environments often differ. By using a macro to define the input filename, you do not need to change the path to the file after you port the job to UNIX. You add the macro to ***install-path*/etc/macros.cfg** in both the Windows and UNIX, and set the path appropriately in each.

The etc directory contains the macros.cfg file and a macros subdirectory. The macros subdirectory can contain multiple .cfg files. If one or more of the .cfg files exist in that subdirectory, then they will be read in alphabetical order before the macros.cfg file is read. The last value read becomes the value that is applied.

If your jobs use system and user-created macros, you must create a combined macro file to be able to use the macros in DataFlux Data Management Server. For more information about macros, see the online Help for DataFlux Data Management Studio.

Declare Input and Output Variables for Data Services

Prior to Release 2.2, the macros that were passed into real-time data services were the only ones that could be returned. Also, any macro variable was allowed to be passed to the service, even if it was not being used by the service.

In Release 2.2 and thereafter, input and output variables for real-time data services behave similarly to variables of real-time process services and batch jobs. Specifically, only input variables that are declared in a real-time 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, then an error is 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

About Updates

For each job process, the DataFlux Data Management Server reads configured macros at the beginning of execution. When a macro changes, you can update the macro on the server without having to restart the server, using one of the following procedures.

Update Macros for Process Services, Batch Jobs, and Profile Jobs

For real-time process services, batch jobs, and profile jobs, all of which are executed by separate instances of the DFWFPROC process:

1. In the Data Management Servers tree in Studio, select a server by name.
2. Right-click on the server name and select **Unload idle processes** from the drop-down menu.

Unloading idle processes also updates macros for all subsequent instances of the DFWFPROC process.

Update Macros for Real-Time Data Services

For real-time data services, all of which are executed by separate instances of the DFWSVC process:

1. In the Data Management Servers tree in Studio, expand a server by name.
2. Select the **Real-Time Data Services** folder under.
3. In the right pane click the **Loaded Processes** tab.
4. Select all of the processes under **Process ID**, and then click one of two buttons depending on the status of the job: **Unload Process When Idle**, or **Unload Process**:

Terminate Real-Time Services

Overview

You (or your client) can submit macros or input variables in SOAP commands to terminate real-time data services and real-time process services. You submit a job run identifier when you request the service. Later, to kill the service, you include the ID in the unload command.

Set a Job ID

When you submit a job run request with ArchitectService or WorkFlowService, you set a job ID, which you can then use to terminate the real-time service. Submit the following key/value pair in a macro (in the varValue element) or as an input variable (in the inputs element):

```
__JOB_METADATA/USER_JOBRUN_ID = your-ID-string
```

You ensure that the value is unique, as necessary, and not NULL. If the value is not unique, then the DataFlux Data Management Server will search the active real-time services terminate the first real-time service with a matching identifier.

Setting a job run identifier provides the job run request with the following two new elements:

svcType
values can be **data** or **process**.

usrJobId
the value is a job run identifier.

Terminate the Real-Time Service

To terminate a real-time data or process service that has job run identifier, include usrJobID and svcType (as needed) in the SOAP command ArchitectServiceUnload.

Manage the DFWSVC Process

Overview

One instance of the DFWSVC process runs one real-time data service. The Data Management Server tracks both idle and active processes. The server also understands whether any service jobs are loaded and waiting to run. When a request for a real-time data service is received from a client, the server first tries to finds an idle DFWSVC process. If one does not exist, then 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.

When an active DFWSVC process is terminated, the DataFlux Data Management Server records the event in the server log. If an idle DFWSVC process terminates, the server logs the event and starts a new process when another request is received.

The maximum run time for data services is set by the configuration option DMSERVER/SOAP/DATA_SVC/MAX_RUNTIME.

Note: When processes are reused too often, performance can degrade. You can specify the amount of reuse in the configuration option POOLING/MAXIMUM_USE, in the server's app.cfg file. After the process has been used the specified number of times, it is terminated.

Unload DFWSVC Processes

Follow these steps to unload one or more real-time data service processes:

1. Open Data Management Studio and connect to the DataFlux Data Management Server.
2. Select the **Real-Time Data Services** folder in the navigation pane.
3. Click the **Loaded Processes** tab.
4. Select one or more real-time data services.

5. Click either **Unload Process When Idle** or **Unload Process**. **Unload Process** unloads the process immediately.

Reference for DFWSVC Configuration Options in dmserver.cfg

Use the following configuration options to manage your DFWSVC processes. These options are specified in ***`install-path`*`/etc/dmserver.cfg`**.

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 time-out. Negative values are ignored.
DMSERVER/SOAP/DATA_SVC/QUEUE	Specifies whether to queue real-time data service requests. The value YES indicates that if all DFWSVC process are busy and no new process are allowed to start, then service requests are placed in a queue. The requests are submitted to DFWSVC processes as processes become available. The default value NO returns an error to the requesting client when the preceding conditions apply.
DMSERVER/SOAP/DATA_SVC/MAX_NUM	Specifies the maximum number of real-time data services that the SOAP server is allowed to run simultaneously. The default value is 10. If a new service request would exceed the set limit, and if the request queue is not enabled, then an error message is sent to the requesting client. This option applies to the SOAP server, meaning that 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 that the WLP server is allowed to run simultaneously. The default value is 10. If a new service request exceeds this limit, an error message is returned to the requesting client. This option applies to the WLP server, which processes service requests from WLP clients. This option does not apply to the SOAP server or requests from SOAP clients.

Configuration Option	Description
DMSERVER/SOAP/DATA_SVC/MAX_ERRS	Specifies the maximum number of service errors that can occur in a DFWSVC process before the process is terminated. The default value is -1, which indicates that there is no process termination due to service errors.
DMSERVER/SOAP/DATA_SVC/ MAX_REQUESTS	Specifies the maximum number of service requests that a DFWSVC is allowed to handle before it is terminated. The default value is -1, which indicates that processes are not terminated based on the number of service requests received.
DMSERVER/SOAP/DATA_SVC/ MAX_RUNTIME	Specifies the number of seconds to allow a real-time data service to produce output data or an error. If a data service does not produce a response within the specified number of seconds, the server terminates the corresponding DFWSVC process. The server then sends a SOAP fault message to the client. The default value for this option is 0 (zero), which indicates that a real-time data service can run indefinitely. Negative values are ignored. Note that the actual time to terminate the process can differ from the value of this option because fractional values are rounded up. An option value of 1.5 can result in a physical duration of 2 seconds before the service is terminated.

Reference for DFWSVC Configuration Options in service.cfg

Use the following configuration options to manage your DFWSVC processes. These options are provided in the file ***install-path*/etc/service.cfg**. When you set these configurations, consider that similar configuration options in the DataFlux Data Management Server's app.cfg configuration file are also applied. The options in app.cfg are applied first, before the configuration options are applied from service.cfg.

Configuration Option	Description
DATASVC/ IGNORE_DECLARED_VARS	The value YES causes DFWSVC to ignore all input and output variables that are declared in the real-time data service. Use this value to run real-time data services that were created prior to DataFlux Data Management Server 2.2. All input macros are passed into the data service and all final macro values are returned to the client along with the output data. The default value NO causes DFWSVC to allow in only the input variables that are declared on the job. If any other input variables are passed in, the real-time data service terminates and returns an error to the client. Only the output variables declared on the job will be returned along with the output data from the service.
DATASVC/ THREAD_STACK_SIZE	This option sets the stack size, in bytes, for each thread of DFWSVC in the UNIX and Linux operating environments. The default value is 1MB. This option is ignored in the Windows operating environment.
DATASVC/ MAX_OUTPUT_BYTES	This option controls the maximum amount of data, in bytes, that the DFWSVC process is allowed to return to the SOAP client. The default value is 128MB (134217728 bytes). If the output data exceeds the limit, the real-time data service is terminated. The error message *Output data size limit exceeded* is logged in the log files of DFWSVC and DataFlux Data Management Server. The SOAP client also receives the error message. A value of 0 or less disables the examination of the output data size. *Note:* This option applies only when DFWSVC is invoked by a SOAP client request. The option is ignored when DFWSVC is invoked by a WLP client request.

Manage the DFWFPROC Process

Overview

The DFWFPROC process runs real-time process services, batch jobs, and profile jobs. Process services are handled independently of batch and profile jobs, by a pooler. The DataFlux Data Management Server requests a new process from the pooler. The server then sends the process the service name and input parameters so that the process can load and run the service.

For batch and profile jobs, the DataFlux Data Management Server starts a new DFWFPROC process and assigns the job to that process. Log entries record job status:

start, running, finished successfully, or terminated due to error. This information is displayed in the Monitor folder in DataFlux Data Management Studio.

You can also configure the DataFlux Data Management Server to collect job run statistics for batch and profile jobs. The statistics are parsed from job run log files by the SAS Job Monitor (in SAS Environment Manager.) The Job Monitor can examine log entries that are collected during job runs. To learn more about using the SAS Job Monitor, see "Collect Job Status Information with the SAS Job Monitor".

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

Limit the Number of Jobs and Queue Job Run Requests

To manage server performance, you can use DMSERVER/MAX_JOB_NUM to control the number of jobs (batch and profile) that can run simultaneously. When the maximum number of active jobs is reached, you can choose to refuse additional job run requests, or to enqueue job run requests using DMSERVER/JOBS_QUEUE. Using the queue, the server honors all job run requests in the order in which they were received, as the number of active jobs drops below the specified limit. By default, the server accepts all job requests and does not use a queue.

In DataFlux Data Management Studio, queued job run requests are displayed in the job's Run History tab. To display the **Run History** tab, you expand the DataFlux Data Management Servers riser, expand the server's entry in the tree list, and click the job. In the **Run History** tab, the **Status** column displays **Queued**. When you right-click a queued job in the **Run History** tab, the only action that does not generate an error message is **Stop**. Stopping the job removes the job run request from the queue.

Similarly, when your SOAP clients submit SOAP commands for enqueued jobs, SOAP Fault messages are returned for all commands except those that stop the job.

Unload Idle DFWFPROC Processes

You can unload idle DFWFPROC processes for real-time process services, batch jobs, and profile jobs using the SOAP command UnloadProcesses, as described in "Apply SOAP Commands and WSDL Options". To unload processes manually, follow these steps:

1. Open Data Management Studio and open the DataFlux Data Management Servers window.
2. In the Data Management Servers window, locate the action menu.
3. Select **Unload Idle Loaded Processes**.

Reference for DFWFPROC Configuration Options

Use the following configuration options to manage your DFWSVC processes. These options are specified in ***install-path*/etc/dmserver.cfg**.

Configuration Option	Description
DMSERVER/SOAP/PROC_SVC/MAX_NUM	Specifies the maximum number of real-time process services that are allowed to run simultaneously. The default value is 10. If a new service request exceeds the limit, then an error message is displayed and the new service is not executed.
DMSERVER/JOBS_MAX_NUM	Specifies the maximum number of batch jobs and profile jobs that are allowed to run simultaneously. The default value is 10. If a new job request exceeds the limit, and if DMSERVER/JOBS=YES, then the job run request is placed in a queue. If DMSERVER/JOBS=NO, then an error message is displayed and the new job is not executed.
DMSERVER/JOBS_QUEUE	Allows or denies the server the ability to queue batch job run requests when the number of running jobs exceeds the value of the DMSERVER/JOBS_MAX_NUM option. A value of YES queues job run requests and sends clients a success response, as if the job had begun to run. As running jobs finish or are canceled, queued jobs run and the queue is cleared. This option does not apply to profile jobs. The default value is NO.
DMSERVER/SOAP/DATA_SVC/ JOB_COUNT_MIN	Specifies the minimum number of instances of a given real-time service that remain loaded. When the number of jobs reaches the minimum limit, the Data Management Server halts the unloading of instances of the specified job. The value of this option is of the form *min-count*:*service-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 limit is reached, the DataFlux Data Management Server will not load any more instances of that service job. The value of this option is of the form ***max-count*:*service-file-name***.
DMSERVER/JOBS_ROOT_PATH	Specifies the location of the root directory for the job and service subdirectories. The default object root directory is ***install-path*\var**. The subdirectories for jobs and services are: **data services**, **process services**, and **batch jobs**.

Run Jobs with the dmpexec Command

Overview

You can run execute jobs on the DataFlux Data Management Server with the command ***install-path*/bin/dmpexec**.

dmpexec Options

The dmpexec command accepts the following options:

Option	Purpose
`-c filename`	Reads a configuration file to set option values that are specific to the job or command, including the authentication option. (See the -a option.)
`-j file`	Executes the job in the specified file.
`-l path-filename`	Writes job run log messages to a file. Specify different log files for each job run, The path value is absolute. It is not affected by the values of any configuration option.
`-i key=value`	Sets the input variable *key* to a value before running the job.
`-o key=value`	Sets a server option to a value.
`-b key=value`	Sets a job option to a value.
`-a`	Authenticates the user who is executing dmpexec using a Metadata Server or Authentication Server. This option is required for domain-enabled connections. To successfully authenticate, you need to specify options that specify the authenticating server, user name, and password.

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 DataFlux Data Management Server requires three server configuration options. The following configuration options specify the location of the authenticating server, a user name that is registered on the authenticating server, and the password that validates the user name:

BASE/AUTH_SERVER_LOC=*network-path*:*port* defines how to connect to the authenticating server.

BASE/AUTH_SERVER_USER=*user-name* specifies the user name that will be authenticated by the specified server.

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

The authenticating server can be a Metadata Server or an Authentication Server.

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 for the dmpexec Command

The dmpexec command returns the following status codes:

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 terminated
4	Job has finished with errors: job was canceled

Configure Jobs and Services

Overview

Jobs and services are configured using the following configuration files, all of which are stored in ***install-path*/etc**:

app.cfg
: Specifies options that determine how job nodes interface with the resources on the Data Management Server. Options in app.cfg specify how job nodes send e-mail, use a Quality Knowledge Base, and access address verification software. Most of these options are commented-out by default. They are enabled only when your jobs need to use a particular resource.

  Real-time data services, real-time process services, batch jobs, and profile jobs are all developed and tested in DataFlux Data Management Studio. When you upload those jobs to DataFlux Data Management Server, the job execution environment has to enable the same configuration options that were used to develop and test those jobs. For this reason, the options that are enabled on the Data Management Server should be similar to the options that are enabled in DataFlux Data Management Studio. Option values differ primarily when they reference storage locations.

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

service.cfg
: Specifies options that apply to real-time data services and real-time process services. This file currently supports one option, BASE/LOGCONFIG_PATH, which specifies the path to the log file directory that is used by service jobs.

batch.cfg
: Specifies options that apply to batch jobs. This file provides an alternate value for the BASE/LOGCONFIG_PATH option.

macros.cfg
: Specifies options (none by default) and macros that apply to all jobs and real-time services. For information about using macros, see "Define Macros".

Options are set by order of precedence, starting in the job's advanced properties. If an option is not specified in the job, then the server checks for a value in macros.cfg, followed by either service.cfg or batch.cfg. If no options are specified, then the default value is retained.

Grant Job Permissions

You can configure user permissions for batch jobs using the DataFlux Data Management Studio administrative interface in DataFlux Data Management Studio. 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

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.

Configure Storage for Temporary Jobs

The business rules monitor creates and executes temporary jobs. Those jobs are normally kept in memory and are not stored on disk. When a directory is specified for temporary jobs, the Monitor stores 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.

Support for Remote-Access Clients

Remote-access clients are published applications (rather than streamed applications) that are run out of software such as Citrix or Microsoft RemoteApps. Your client applications, and DataFlux Data Management Studio, can be run as remote-access clients.

Remote-access clients require additional support to ensure that the cancellation of jobs results in the termination of all remote child processes. To effectively cancel remote processes, set the following option in ***install-path*/etc/app.cfg**:

```
BASE/MAINTAIN_GROUP=YES
```

If you do not set the MAINTAIN_GROUP option, then the cancellation of jobs can allow child processes to persist on remote-access clients. These rogue processes can become associated with a new group or job.

If you set the MAINTAIN_GROUP, and if remote child processes persist, then you might have to restart the remote-access client to terminate the processes.

Resolve Out-of-Memory Errors When Using Sun JVM

You can encounter out-of-memory errors in jobs with a SOAP Request node or an HTTP Request node, when you are using a Sun Java Virtual Machine (JVM). To resolve this error, add the following option to the Java start command that is specified in the app.cfg file:

```
-XX:MaxPermSize=256m -XX:+CMSClassUnloadingEnabled
```

Collect Job Status Information with the SAS Job Monitor

You can collect status information about your job runs using the SAS Job Monitor. The SAS Job Monitor is supplied as part of the SAS Environment Manager software. The SAS Job Monitor collects statistics from job run log files. Job run log files are generated by the Monitor logger on the Data Management Server. The Monitor logger is active and configured on the server by default.

By default, the Job Monitor collects statistics at the end of the job run. To collect statistics at specified intervals *during* job runs, add the configuration option BASE/MONITOR_FREQUENCY to the file ***install-path*/etc/app.cfg**. The default value of this option is -1. To collect statistics during the job run, set an option value that determines the time interval in milliseconds between the generation of log entries. Experiment with different option values to achieve desired results.

To store job run logs in a separate directory, specify a directory path as the value of the option DMSERVER/JOB_LOGS_DIR, in the file ***install-path*/etc/dmserver.cfg**. By default, job run logs are stored in the same directory as all other job logs and job files, as specified by the value of DMSERVER/WORK_ROOT_PATH. The default directory can become large and difficult to navigate. If you specify DMSERVER/JOB_LOGS_DIR, the value of that option will be set when you start the server. At start time, the value of the option is recorded in the server log file.

If you need to change the encoding of your job logs, set the option BASE/JOB_LOG_ENCODING in the DataFlux Data Management Server's app.cfg file. The BASE/JOB_LOG_ENCODING option specifies the encoding that is used in the job log. By default, the log is written in the encoding associated with the locale of the process that executes the job. For English-speaking organizations, this might be LATIN-1 or UTF-8. If a log entry contains characters that cannot be represented in the specified encoding, then the log entry is not written to the log file.

For additional information about the collection of job statistics, see the *Data Management Studio Installation and Configuration Guide*, the *Data Management Studio User's Guide*, and the Help for the SAS Job Monitor.

About the Repository

A repository is a directory that stores a database file or database connection information. You run jobs on the DataFlux Data Management Server to access the data in the repository 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, then 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 Unified Database connection parameters. 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 ***install-path*\etc\dmserver.cfg**, and then restart the DataFlux Data Management Server.

Create a Repository

The repository can be configured as a directory of files, a database file (such as a SQLite file), or a database connection.

Follow these steps to create a repository of any type:

1. In DataFlux Data Management Studio, click the **Administration** riser bar.
2. In the **Repository Definitions** pane, click the **New** button.
3. In the New Repository Definition dialog box, enter the name of your repository in the **Name** field.
4. In the Data Storage section of the dialog box, if you want to configure your repository as a database file, then select **Database file**.
5. If you want to configure your repository as a database connection, then select **Database connection**.
6. If you want to configure your repository as a directory of files, click the **Browse** button and specify a physical path. The path must be accessible to everyone who needs access to the real-time data services, real-time process services, .sas files, and entity resolution output files (.sri) that will be stored in that location.
7. Select or deselect **Connect to repository at startup**, and then uncheck **Private**. If you select **Private**, then the repository is stored to the local host and it cannot be viewed by others.
8. Click **OK** to create the new repository. The default location of the new .rcf repository file in Studio is: ***install-path*\etc\repositories**.
9. Follow these steps if your repository will consist of files or a database file:

- Copy the new .rcf file from the DataFlux Data Management Studio host to the same directory on the Data Management Server host.
- Copy the repository files from the DataFlux Data Management Studio host to a convenient location on the Data Management Server host.
- Edit the .rcf file to point to the location of the files on the DataFlux Data Management Server host.

10. Follow these steps if your repository will consist of a database connection:
    - Copy the new .rcf file from the DataFlux Data Management Studio host to the same directory on the Data Management Server host.
    - Create an ODBC data connection with the same name that is used in the .rcf file.

Troubleshoot Jobs and Services

Overview

If your job or service experiences any of the following symptoms, refer to the following resolutions.

Server Processes (DFWSVC or DFWFPROC) Fail to Start, or Out of Memory Error in Windows When Launching Server Processes

Windows displays the following error message:

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

The Data Management Server log file might also display one of the following messages:

```
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
getprocess, errorCode=2 (Process
'HOST:ADDR' exited unexpectedly.)

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

It is possible for the Windows event log to not contain entries for DFWSVC and DFWFPROC, even when the DataFlux Data Management Server logs contain one or more entries. This symptom often indicates that the failure to start processes is caused by Windows running too many internal processes. The Data Management Server cannot start new processes.

The log discrepancy occurs when Windows 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 DataFlux Data Management Server. If the errors persist, you might need to make a minor change in the Windows registry to increase the SharedSection parameter of the SubSystems key in HKEY_LOCAL_MACHINE. For additional information, see the following Microsoft Support articles:

- **"Out of Memory" error message appears when you have a large number of programs running**
- **User32.dll or Kernel32.dll fails to initialize**
- **Unexpected behavior occurs when you run many processes on a computer running SQL Server**

Required openSSL DLLs Were Not Found

This error message indicates that the DLLS for openSSL were placed in a directory other than /bin during installation. Copy the DLLS to the /bin directory and restart the server.

The Repository Is Newer Than This Client

This error message indicates that someone at your site has upgraded the repository on the Data Management Server. Install a new version of DataFlux Data Management Studio to use the new repository.

SQL Lookup Job Fails on a UNIX or Linux System Using the Driver for BASE

The Driver for BASE does not allow data sets to be created that cannot be read by SAS. If you have Driver for SAS files that contain letters that cannot be accessed in the UNIX or Linux operating environments, then you will need to rename the file to all-lowercase. Other files that contain mixed case or uppercase letters might 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 might be named lookupsource. In jobs, you can reference LOOKUPSOURCE, lookupsource, or LookUPSoUrCe, just to name a few.

When Opening a Job Log: SOAP-ENV:Client:UNKNOWN Error (or Time-out)

This error occurs on some configurations of Microsoft Windows Server 2003, when the log file exceeds 32KB. A workaround for this problem is to set the following configuration value in the dmserver.cfg file:

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

This error and this resolution apply only when the host of your Data Management Server is running Windows Server 2003.

Error Occurs in an Address Verification Job on Linux

The following error message content indicates that the job is attempting to use an unsupported version of Address Doctor:

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

To resolve this error, edit the job to use the latest Address Verification node, which uses the latest version of the Address Doctor software.

Blue Fusion Cannot Process New Quality Knowledge Base Definitions

This error occurs when the currently loaded QKB uses definitions that are newer than those that are supported by the current version of Blue Fusion. By default, Blue Fusion attempts to load the definitions and issues warnings before loading them. If the definitions include instructions that Blue Fusion cannot process, the instructions are ignored and an error is displayed.

The QKB/ALLOW_INCOMPAT option can be used to specify whether to allow incompatible QBK definitions to be processed by Blue Fusion. The option is defined in the app.cfg file; it enables you to choose to either stop processing or allow the incompatibility and continue processing the definitions.

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.
```

To resolve this error, ensure that the name of the scheme is entered correctly, as it is case sensitive. Also ensure that the Quality Knowledge Base (QKB) you are using is an exact copy of the QKB used when the job was created in DataFlux Data Management Studio.

To copy the QKB from Windows to UNIX or Linux, use FTP or Samba mappings. After you copy the QKB, restart the DataFlux Data Management Server and run the job again. In UNIX and Linux, change the scheme name (in the scheme directory of the QKB) as needed to use all lowercase letters.

Customize the Server's WSDL File

The Data Management Server's client library is available in Java and C. You can customize the file to suit your environment using the Web Service Definition Language (WSDL) file. You can access the WSDL file at the following path or by entering the following URL into a web browser: contains the descriptions of the available web services. You can access the WSDL file directly, or you can use the following URL in a web browser:

install-path/share/arch.wsdl

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

In the WSDL file, the value of the **SOAP:address** location reflects the local server's host name and port number. Using an XML editor, you can update the **SOAP:address**

location to reflect the host name and port number of any DataFlux Data Management Server. One note of caution, please do not edit any other values in the arch.wsdl file. For example:

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

Customize the WSDL File for Java

To customize arch.wsdl for Java, use a mapping tool, such as wscompile, to generate stubs and build classes that wrap the DataFlux Data Management Server interface. In the examples below, using wscompile, the WSDL file is imported and the stubs are generated using 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);
```

Customize the WSDL File for C#

To customize your arch.wsdl file for a C# environment, import a web reference into your project. This builds the object that is 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;
```

Chapter 7

Configuration Option Reference

Configuration Options Overview

This chapter provides detailed information for the configuration options that are maintained in the file ***install-path*\etc\dmserver.cfg**. Edit the configuration file to change option values, and then restart the server to apply your changes.

This chapter documents only the subset of the options in dmserver.cfg that pertain directly to the DataFlux Data Management Server. For information about options in dmserver.cfg that are not documented in this chapter, refer to the Configuration Options Reference in the *DataFlux Data Management Studio Installation and Configuration Guide*.

The Data Management Server also uses configuration options that are provided in the file ***install-path*\etc\app.cfg**. Reference material for the configuration options in the app.cfg file is also provided in the *DataFlux Data Management Studio Installation and Configuration Guide*.

Configuration options in dmserver.cfg and app.cfg are generally maintained in parallel between a given DataFlux Data Management Server and the instances of Data Management Studio that upload jobs to that server. Maintaining parallel configurations helps ensure that jobs that are tested in DataFlux Data Management Studio will run after they are uploaded to DataFlux Data Management Server.

Configuration Options Reference for dmserver.cfg

Configuration Option	Description
BASE/AUTH_SERVER_PASS	Specifies a default password that is submitted to the authenticating server (SAS Metadata Server or DataFlux Authentication Server) when you use issue the command **`dmpexec -a`**. This option is overridden when the dmpexec command provides a job-specific value. For more information, see "Run Jobs with the dmpexec Command".
BASE/AUTH_SERVER_USER	Specifies a default user name that is submitted to the authenticating server (SAS Metadata Server or DataFlux Authentication Server) when you use issue the command **`dmpexec -a`**.
DMSERVER/CHILD/ LISTEN_HOST	Specifies the host name or IP address to which the DataFlux Data Management Server must bind for DFWSVC child process connections. The default value is **`localhost`**. For more information about binding to a host name or IP address, see the option DMSERVER/SOAP/LISTEN_HOST.
DMSERVER/CHILD/ LISTEN_PORT	Specifies the port on which the DataFlux 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 about 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, access is enabled for all IP addresses. As shown in . $$$ Control Access by IP Address, a non-default value allows or denies access to specified IP addresses.
DMSERVER/IPACC/ NOSECURITY	Allows or denies to specified IP addresses the ability to bypass all security checks on the DataFlux Data Management Server. This option is disabled by default.
DMSERVER/IPACC/ POST_DELETE	Controls by IP address a client's ability to submit SOAP post and delete requests. These controls restrict uploads and deletions of objects, such as jobs and services. This option is disabled by default.

Configuration Option	Description
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 and statistics files. If the DMSERVER/ JOBS_KEEP_HISTORY=NO, then a history record is also deleted from history database. The default value of the MAXAGE option is -1, which specifies that history items are 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/JOB_LOGS_DIR	Specifies the path and directory that is used to store the log files that are generated for each run of a batch job or a profile job. This option enables you to separate your job run logs from other logs and other files that are generated by job runs. Separating the job run logs makes it easier to see the files that are used to collect job run statistics. Statistics can be collected by the Job Monitor plug-in SAS Environment Manager. The default value of this option is specified by DMSERVER/WORK_ROOT_PATH, which specifies the storage location for all job-related logs and files.
DMSERVER/JOBS_MAX_NUM	Specifies the maximum number of batch and profile jobs that the DataFlux Data Management Server runs simultaneously. (Batch and profile jobs are counted against the same pool). The default value is 10. If a new job request exceeds the limit, and if DMSERVER/ JOBS=YES, then the job request is placed in a queue. If DMSERVER/JOBS=NO, then an error message is displayed and the new job is not executed.
DMSERVER/JOBS_NO_STATE	Allows or denies batch and profile jobs the ability to generate state files for their runs. The default value NO, which means that jobs are allowed to generate state files. A value of YES prevents the generation of state files. If your job is denied the ability to generate a state file, the server returns the SOAP Fault message State Generation Not Allowed.

Configuration Option	Description
DMSERVER/JOBS_QUEUE	Allows or denies the server the ability to queue batch and profile job run requests when the number of running jobs exceeds the value of the DMSERVER/JOBS_MAX_NUM option. A value of YES queues job run requests and sends clients a success response, as if the job had begun to run. As running jobs finish or are canceled, queued job run and the queue is cleared. This option does not apply to profile jobs. The default value is NO.
DMSERVER/JOBS_ROOT_PATH	Specifies the location of the root directory for the jobs and services subdirectories. The default root directory is ***install-path*\var**. The subdirectories 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 to the client, in response to the getJobLog request. For a 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.

Configuration Option	Description
DMSERVER/NAME	Specifies the name of the metadata definition of the DataFlux Data Management Server that is stored on the SAS Metadata Server. When the DataFlux Data Management Server is started, it uses the name to query the SAS Metadata Server for configuration information. When the option BASE/AUTH_SERVER_LOC in app.cfg identifies a SAS Metadata Server, the DataFlux Data Management Server retrieves and sets the following values: • DMSERVER/SOAP/LISTEN_HOST • DMSERVER/SOAP/LISTEN_PORT • DMSERVER/SOAP/SSL • DMSERVER/SECURE If the SAS Metadata Server cannot locate a metadata definition based on the name, then the DataFlux Data Management Server does not start. If any of the preceding options have values in the DataFlux Data Management Server's dmservre.cfg file, then the local values override the values that are supplied in metadata. For this reason, it is recommended that you comment-out these options in dmserver.cfg. To access the named metadata definition on the SAS Metadata Server, one of two conditions must be met. You can ensure that the process owner of the Data Management Server has a user definition on the SAS Metadata Server. Otherwise, the named metadata definition needs to be available to the PUBLIC group. This option is ignored when BASE/AUTH_SERVER_LOC identifies a DataFlux Authentication Server rather than a SAS Metadata Server. This option is specified by default when the DataFlux Data Management Server is installed as part of SAS Visual Process Orchestration.
DMSERVER/NO_WORK_SUBDIRS	Specifies whether to create separate log subdirectories. The default value is NO, which means that all log files are created in subdirectories under the default directory, server_logs, or an alternate directory specified in the DMSERVER/WORK_ROOT_PATH option. The value YES should be applied only in special cases, as it creates numerous log files in a single directory. This single directory 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.

Configuration Option	Description
DMSERVER/SECURE	Enables or disables authorization and authentication on the DataFlux Data Management Server. This option also enables the use of extended encryption algorithms for stored passwords and IOM (TCP/IP) communication. The default value NO specifies that SECURE configuration options are ignored. The value YES specifies that other configuration options are required to properly secure the server, including BASE/AUTH_SERVER_LOC. When using a SAS Metadata Server for security, the value of this option is retrieved from metadata when you start the DataFlux Data Management Server. At start time, if the dmserver.cfg file contains a value for this option, then the local value overrides the metadata value. For this reason, it is recommended that you not enable this option in dmserver.cfg when using a SAS Metadata Server. For further information, see DMSERVER/NAME, DMSERVER/SOAP/SSL, and "Configure SSL and AES".
DMSERVER/SECURE/DEFAULT_ACE_PUBLIC	Allows or denies the DataFlux Data Management Server the ability to create a default access control entry for the PUBLIC group. The server can create a default ACE when it creates a default access control list (ACL) for an object. The server creates a default ACL when an object is uploaded to the server. The server also creates a default ACL when an existing object on the 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 valid values for this option are ALLOW and DENY. Any other value is treated as INHERIT.
DMSERVER/SECURE/DEFAULT_ACE_USERS	Allows or denies the DataFlux Data Management Server the ability to create a default access control entry for the USERS group. The server can create a default ACE when it creates a default access control list (ACL) for an object. The server creates a default ACL when an object is uploaded to the server. The server also creates a default ACL when an existing object on the server is accessed and a corresponding ACL does not exist. By default, an ACE is created for the USERS group, which implies INHERIT access for that group. The valid values for this option are ALLOW and DENY. Any other value is treated as INHERIT.

Configuration Option	Description
DMSERVER/SECURE/ GRP_ADMIN	Specifies the name of the DataFlux Data Management Server administrator group. If this option is defined, the group must exist on the authenticating server (SAS Metadata Server or DataFlux Authentication Server.) An error message is generated at server invocation when the following conditions exist: • DMSERVER/SECURE/GRP_ADMIN is not defined • the group is not defined • DMSERVER/SECURE=YES
DMSERVER/SECURE/ GRP_ALLOW	Specifies the name of a non-administrative group, to provide access to the Data Management Server for a few users and to exclude the rest. If this option is not set, all users are allowed in accordance with permissions and other configured options. The group must be defined by the same name on the authenticating server (SAS Metadata Server or DataFlux Authentication Server) before the group name can be applied as the value of this option..
DMSERVER/SECURE/GRP_DENY	Specifies the name of a non-administrative group, to provide access to the Data Management Server for most users and to exclude a few users. If this option is not set, all users are allowed in accordance with permissions and other configured options. The group must be defined by the same name on the authenticating server (SAS Metadata Server or DataFlux Authentication Server) before the name can be applied as the value of this option..
DMSERVER/SOAP/ CONNS_BACKLOG	Specifies the maximum size of the connection request queue. The queue size limit enables the SOAP server to refuse connection requests when it can no longer process them within an acceptable period of time. 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 time-out. Negative 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 any given time. When this limit is reached, the DataFlux Data Management Server will not load any more instances of that service job. The option value syntax is **`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. When this limit is reached, the DataFlux Data Management Server will not unload any more instances of that service job. The option value syntax is **`count:job_file_name`**.

Configuration Option	Description
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 that the SOAP server is allowed to run simultaneously. The default is 10. If a new service request would exceed the limit, and if a queue is not enabled, then an error message is displayed. This option applies to the SOAP server, meaning that the service requests are coming from SOAP clients. This option does not apply to the WLP server or to requests from WLP clients.
DMSERVER/SOAP/DATA_SVC/MAX_REQUESTS	Specifies the maximum number of service requests that a given instance of the DFWSVC process is allowed to handle before that instance of the DFWSVC process is forced to terminate. The default is -1, meaning that no limit is enforced..
DMSERVER/SOAP/DATA_SVC/MAX_RUNTIME	Specifies the maximum number of seconds that a data service is allowed to run a job and produce a response (output data or an error). If a data service does not produce a response within the time limit, then the corresponding instance of the DFWSVC process is terminated. The client receives a SOAP Fault message. The default value is zero, which means that no time-out occurs. Negative values are ignored. Note that the time-out can vary by one or two seconds due to the rounding up of counts less than a second.
DMSERVER/SOAP/DATA_SVC/PRELOAD	Specifies the number of instances of specified services that the DataFlux Data Management Server preloads during start-up. The option value syntax consists of one or more blank-separated instances of **`count:name-of-service`**.This option can be used with the option 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 preload at server initialization a specified number of instances of all real-time data services, including those found in subdirectories. The value of the option must be an integer greater than zero. Otherwise, the option is ignored. This option can be used with DMSERVER/SOAP/DATA_SVC/PRELOAD.

Configuration Option	Description
DMSERVER/SOAP/DATA_SVC/PRELOAD_DURING_RUN	Allows or denies the DataFlux Data Management Server the ability to start a separate thread to preload services and accept SOAP requests immediately during preload. By default, the value NO specifies that the server preloads services before it accepts SOAP requests. The value YES specifies that the server start a separate thread to preload services and accept SOAP request simultaneously. If the server stops while the preload thread is running, that thread is terminated.
DMSERVER/SOAP/DATA_SVC/QUEUE	Allows or denies the DataFlux Data Management Server the ability to queue requests for real-time data services. The default value NO indicates that service request generate a SOAP Fault error when all instances of the DFWSVC process are busy. The value YES indicates that the DataFlux Data Management Server queues service requests, and processes those requests when instances of the DFWSVC process become available.
DMSERVER/SOAP/IGNORE_NS	Allows or denies the DataFlux Data Management Server the ability to ignore namespaces in SOAP requests. The default value NO specifies that NULL values in input data fields are passed to jobs as empty strings. The value YES causes the server to ignore namespaces in SOAP requests. This allows the SOAP server to preserve NULL values when receiving input data instead of converting the values to empty strings.
DMSERVER/SOAP/LISTEN_PORT	Specifies the port on which the SOAP server listens for connections. The default value is 21036 when you use a DataFlux Authentication Server for security. When you use a SAS Metadata Server for security, the DataFlux Data Management Server uses the DMSERVER/NAME option to retrieve from metadata the values of three configuration options, including LISTEN_PORT. If the SAS Metadata Server does not return a value for LISTEN_PORT, then the DataFlux Data Management Server does not start. If a value is returned, and if dmserver.cfg also contains a value for LISTEN_PORT, then the local value overrides the metadata value. For this reason, it is recommended that you not set LISTEN_PORT in dmserver.cfg when using a SAS Metadata Server. For further information, see DMSERVER/NAME and DMSERVER/SOAP/SSL.
DMSERVER/SOAP/LOG_PACKETS	Enables or disables the generation of the Specifies whether to generate the _PACKETS_ log file. A value of YES specifies that 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.

Configuration Option	Description
DMSERVER/SOAP/PROC_SVC/MAX_NUM	Specifies the maximum number of real-time process services that the DataFlux Data Management Server runs simultaneously. The default is ten. If a new service request exceeds this limit, an error message is displayed.
DMSERVER/SOAP/RDWR_TIMEOUT	Specifies the time-out value for Read and Write operations on a socket. Set a positive value to seconds, a negative value to microseconds, and no time-out to 0. If a non-0 value is set, a time-out occurs if no data can be sent or received within the specified time limit. The time-out count begins after the server initiates a send or receive operation over the socket. When a time-out occurs, a SOAP_EOF error is returned. The default value is zero seconds.
DMSERVER/SOAP/RETURN_NULLS	Specifies how real-time data service return null values in output fields. The default value NO specifies that empty strings are returned. The value YES specifies that NULLs are returned.
DMSERVER/SOAP/SSL	Enables or disables the use of SSL to protect SOAP communication. Encryption for SSL is also enabled, using algorithms that exceed the default SASProprietary. The default value of this option is NO. Specify a value of YES to enables the following SSL configuration options: • DMSERVER/SOAP/SSL/CA_CERT_FILE • DMSERVER/SOAP/SSL/CA_CERT_PATH • DMSERVER/SOAP/SSL/KEY_FILE • DMSERVER/SOAP/SSL/KEY_PASSWD When you use a SAS Metadata Server for security, the value of the DMSERVER/SOAP/SSL option is retrieved from metadata when you start the DataFlux Data Management Server. At start time, if the dmserver.cfg file contains a value for this option, then the local value overrides the metadata value. For this reason, it is recommended that you not specify a value for this option in dmserver.cfg. When you use a DataFlux Authentication Server for security, specify a value of YES in dmserver.cfg to enable SSL. For further information, see DMSERVER/NAME, DMSERVER/SECUREand “Configure SSL and AES”.
DMSERVER/SOAP/SSL/CA_CERT_FILE	Specifies the file where the Certificates Authority stores trusted certificates. If this option is not needed, then comment it out.

Configuration Option	Description
DMSERVER/SOAP/SSL/ CA_CERT_PATH	Specifies the path to the directory where trusted certificates are stored. If this option is not needed, then comment it out.
DMSERVER/SOAP/SSL/ KEY_FILE	Specifies the path to the key file that is required when the SOAP server authenticates clients. If this option is not used, then 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, then comment-out this option. The value of this option must be encrypted for the option to be recognized. To encrypt passwords, see "Encrypt Passwords for DSNs and SSL".
DMSERVER/SOAP/WSDL	Allows or denies the DataFlux Data Management Server the ability to load WSDLs when the server is initialized. Specify a value of YES to load existing WSDLs at start-up, recognize jobs that are started by runSVC requests (if matching WSDLs exist,) and to recognize other WSDL configuration options. The default value NO specifies that WSDLs are not used, runSVC requests are ignored, and other WSDL configuration option are ignored.
DMSERVER/SOAP/WSDL/GEN	Enables or disables the generation of run-time WSDLs. The default value NO specifies that runSVC requests for run-time WSDL jobs will generate a SOAP Fault. Requests to upload jobs with run-time WSDLs are also ignored. The only WSDL jobs that run are those that have WSDLs that are preloaded at the invocation of the DataFlux Data Management Server. The value SINGLE enables the generation of a single WSDL for each postJob request or genWSDL request for a single file. The value MULTIPLE enables generation of multiple WSDLs for genWSDL requests that apply to multiple job files or to 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 note that a request to generate WSDLs for all jobs under the root directory can cause a severe degradation in server performance. Specify the value MULTIPLE with caution.

Configuration Option	Description
DMSERVER/SOAP/WSDL/GEN_ON_GET	Specifies how the Data Management Server generates WSDLs for jobs that include run-time WSDLs. The value NO specifies that an error message is generated in response HTTP getWSDL requests when: • a WSDL does not exist on the server, or • the WSDL does exist on the server, but the **mod.time** timestamp in the server WSDL differs from the same value in the client WSDL. The difference in **mod.time** values indicates that the client WSDL differs from the corresponding server WSDL. The value YES indicates that the server responds to the preceding conditions by generating a new WSDL and sending the new WSDL to the client. This option is valid only when the value of DMSERVER/SOAP/WSDL/GEN is SINGLE or MULTIPLE.
DMSERVER/SOAP/WSDL/RUN_IGNORE_MTIME	Allows or denies the server the ability to detect differences between the **mod.time** timestamps in the DataFlux Data Management Server WSDL and in the WSDL of the client that submits a service request. The default value NO indicates that the server responds to **mod.time** differences by sending the client the SOAP Fault message **Job Has Changed**. This message indicates to the client that the client needs to request the server to generate a new WSDL and send the new WSDL to the client. The client can then update its version of the WSDL. The value YES specifies that the server does not compare **mod.time** values. Service requests are passed to a WLP process for execution. Note that this is the behavior of service requests based on generic WSDL.
DMSERVER/THREADS/COUNT_MAX	Specifies the maximum number of threads that can be started by the DataFlux Data Management Server. 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 that can be kept open by the DataFlux Data Management Server. The default value of zero indicates that threads are terminated when they become idle. If that thread is needed again, it is restarted.
DMSERVER/THREADS/IDLE_TIMEOUT	Specifies the number of microseconds before a thread is flagged as idle after the thread stops doing work. The default of zero indicates that threads are initially flagged as idle.

Configuration Option	Description
DMSERVER/WLP	Enables or disables the execution of the WLP server. The default value NO specifies that the WLP server is bypassed at the invocation of the Data Management Server. WLP clients then cannot connect to the DataFlux Data Management Server. (SOAP clients can still connect to the server.) The value YES specifies that the WLP server starts and listens at its assigned port. The Data Management Server log file will contain status entries for the WLP server.
DMSERVER/WLP/DATA_SVC/MAX_NUM	Specifies the maximum number of real-time data services that the WLP server is allowed to run simultaneously. The default is 10. If a new service request exceeds this limit, an error message is sent to the requesting WLP client. This option does not apply to the SOAP server or requests coming from SOAP clients.
DMSERVER/WLP/LISTEN_HOST	Specifies a host name or IP address to which the WLP server must bind. This specific binding prevents the WLP server from responding to requests that are sent to its port with a different host name or IP address. A pair of host names and IP addresses can be used interchangeably. For example, if the option value is **localhost**, then local clients sending requests to 127.0.0.1 will still be heard. However, requests that are sent to a public IP address, to the host name of the machine, or those that come from external clients will not be heard. By default, this option is left blank, so that the WLP server is not bound to a specific host name or IP address. The WLP server responds to all requests that apply to the DataFlux Data Management Server host and to the port of the WLP server.
DMSERVER/WLP/LISTEN_PORT	Specifies the port on which the WLP server listens for requests 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 the DataFlux 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 start-up date and time, as well as the corresponding process ID. The default directory is ***install-path*\var\server_logs**.

Glossary

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 DataFlux 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.

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

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

dfwsvc
A DataFlux 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.

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.

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.

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.

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

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 DataFlux Data Management Studio.

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

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.

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

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

Index