

DataFlux Data Management Server



YOUR DATA.
YOUR BUSINESS.
ONE SOLUTION.



This page is intentionally blank



DataFlux Data Management Server Administrator's Guide

Version 2.1

June 8, 2010

This page is intentionally blank

Contact DataFlux

Corporate Headquarters

DataFlux Corporation

940 NW Cary Parkway, Suite 201
Cary, NC 27513-2792
Toll Free Phone: 877-846-FLUX (3589)
Toll Free Fax: 877-769-FLUX (3589)
Local Phone: 1-919-447-3000
Local Fax: 919-447-3100
Web: <http://www.dataflux.com>

DataFlux United Kingdom

Enterprise House
1-2 Hatfields
London
SE1 9PG
Phone: +44 (0) 20 3176 0025

DataFlux Germany

In der Neckarhelle 162
69118 Heidelberg
Germany
Phone: +49 (0) 6221 4150

DataFlux France

Immeuble Danica B
21, avenue Georges Pompidou
Lyon Cedex 03
69486 Lyon
France
Phone: +33 (0) 4 72 91 31 42

Technical Support

Phone: 919-531-9000
Email: techsupport@dataflux.com
Web: <http://www.dataflux.com/MyDataFlux-Portal>

Documentation Support

Email: docs@dataflux.com

Legal Information

Copyright © 1997 - 2010 DataFlux Corporation LLC, Cary, NC, USA. All Rights Reserved.

DataFlux and all other DataFlux Corporation LLC product or service names are registered trademarks or trademarks of, or licensed to, DataFlux Corporation LLC in the USA and other countries. ® indicates USA registration.

[DataFlux Legal Statements](#)

[DataFlux Solutions and Accelerators Legal Statements](#)

DataFlux Legal Statements

Apache Portable Runtime License Disclosure

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

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

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

Apache/Xerces Copyright Disclosure

The Apache Software License, Version 1.1

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

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

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

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

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

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

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

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

DataDirect Copyright Disclosure

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

Expat Copyright Disclosure

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

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

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

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

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

gSOAP Copyright Disclosure

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

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

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

IBM Copyright Disclosure

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

COPYRIGHT AND PERMISSION NOTICE

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

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

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

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

Microsoft Copyright Disclosure

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

Oracle Copyright Disclosure

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

PCRE Copyright Disclosure

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

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

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

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

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

Red Hat Copyright Disclosure

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

SAS Copyright Disclosure

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

SQLite Copyright Disclosure

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

Sun Microsystems Copyright Disclosure

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

Tele Atlas North American Copyright Disclosure

Portions copyright © 2006 Tele Atlas North American, Inc. All rights reserved. This material is proprietary and the subject of copyright protection and other intellectual property rights owned by or licensed to Tele Atlas North America, Inc. The use of this material is subject to the terms of a license agreement. You will be held liable for any unauthorized copying or disclosure of this material.

USPS Copyright Disclosure

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

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

Solutions and Accelerators Legal Statements

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

Apache

This product may contain software technology licensed from Apache.

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

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

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

Creative Commons Attribution

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

Degrafa

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

Copyright © 2008-2010 Degrafa. All rights reserved.

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

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

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

Google Web Toolkit

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

JDOM Project

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

OpenSymphony

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

Sun Microsystems

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

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

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

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

Java Toolkit

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

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

Table of Contents

Introduction	1
Conventions Used In This Document	1
Reference Documentation.....	2
Minimum Requirements	3
Supported Operating Systems.....	3
Supported Databases	4
Bundled UNIX Drivers	4
Overview	5
What Data Management Server Does	6
Where and How Data Management Server Runs	6
DataFlux Data Management Server Configuration Options	6
Installing Data Management Server	7
Windows	7
Directory Layout for Data Management Server Installation	8
Installing Other DataFlux Products.....	9
Security Considerations.....	9
Data Management Server Components	10
Data Management Server Service.....	10
Multi-threaded Operation.....	11
Data Management Server – Standard.....	12
Data Management Server – Enterprise	13
Data Management Server Jobs	15
Data Management Server Processes	17
Annual Licensing Notification.....	18
Data Management Server Security	19

Enabling Security	19
Security Files	20
Authentication.....	21
Command Permission Bits	22
Authorization	23
Security Management Commands.....	25
Command Permissions and Access Rules	26
IP-Based Security	28
Upgrade Path for Security-Related Objects	29
Configuring DataFlux Data Management Server	31
Data Management Server Configuration Directives	31
Configuring Licensing	38
Configuring a Data Source	40
Configuring the Data Access Component	40
Changing Configuration Settings	41
SOAP Commands and WSDL Types	43
Technical Support	46
Best Practices	46
Frequently Asked Questions.....	47
Troubleshooting	54
Error Messages	56
Appendix A: Code Examples	61
Java.....	61
C++	64
C#.....	67
Appendix B: Security Policy Planning and Examples	72
Security Policy Planning.....	72

Data Management Server Security Examples	72
Appendix C: Using Configuration Settings	75
Using Configuration Settings to Pre-Load Services	75
Using Configuration Settings for Multi-Threaded Operation	77
Glossary	78

Introduction

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

- [Conventions Used in This Book](#)
- [DataFlux Reference Documentation](#)

Conventions Used In This Document

This document uses several conventions for special terms and actions.

Typographical Conventions

The following typographical conventions are used in this document:

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

Syntax Conventions

The following syntax conventions are used in this document:

Syntax	Description
#	The pound # sign at the beginning of example code indicates a comment that is not part of the code.
>	The greater than symbol is used to show a browse path, for example Start > Programs > DataFlux Data Management Server > Administrator's Guide.

Reference Documentation

This document may reference other DataFlux documentation, including:

DataFlux Authentication Server Administrator's Guide

DataFlux Authentication Server User's Guide

DataFlux DataFlux Federation Server Administrator's Guide

DataFlux DataFlux Federation Server User's Guide

DataFlux Data Management Server User's Guide

DataFlux Data Management Server Installation Guide

DataFlux Data Management Studio Installation and Configuration Guide


DataFlux Data Management Studio Online Help

DataFlux Expression Language Reference Guide

DataFlux Quality Knowledge Base Online Help

Minimum Requirements

The following sections list the minimum requirements for a DataFlux® Data Management Server (Data Management Server) installation. This includes the supported operating systems and databases, and the UNIX drivers that are bundled with the server.

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

The following table lists the minimum requirements of the computer on which the Data Management Server is installed:

Requirement	Minimum	Recommended
Platforms ¹	For information, see Supported Operating Systems .	N/A
Processor	For information, see Supported Operating Systems .	N/A
Memory (RAM)	1 GB ²	2 GB per CPU core ²
Disk Space	1 GB for Installation 1 GB for temp space	10 GB for Installation ³ 20 GB for temp space ³

Notes:

1. Other platforms are available. Contact [DataFlux](#) for a complete list.
2. Actual requirements depend on configuration.
3. Verification processes rely on reference databases to verify and correct address data. The size of these reference databases varies. Check with DataFlux for exact size requirements for this component.

Supported Operating Systems

The following is a list of the minimum requirements for supported platforms for a Data Management Server installation. The minimum operating system requirements may be different if you are accessing SAS data sets. In some instances, you may be required to run a more recent version of the operating system, as noted in parentheses:

Platform	Bits	Operating System	Hardware Architecture
AIX®	64	IBM® AIX 5.3	POWER/Power PC®
HP-UX (PA-RISC)	64	HP-UX 11i Version 1.0 (11.11) (SAS: HP-UX 11.23 or later)	PA-RISC 2.0
HP-UX (Itanium)	64	HP-UX 11i Version 2.0 (11.23) (SAS: June 2007 patch bundle)	Itanium® (IA64)

Platform	Bits	Operating System	Hardware Architecture
Linux®	32	Linux 2.6 (glibc 2.3) (SAS: Red Hat Enterprise Linux 4 and above; SuSE Linux Enterprise Server 9 or later)	Intel® Pentium® Pro (i686)
Linux	64	Linux 2.6 (glibc 2.3) (SAS: Red Hat Enterprise Linux 4 and above; SuSE Linux Enterprise Server 9 or later)	AMD AMD64 or Intel EM64T
Solaris™ (SPARC)	64	Sun™ Solaris 9 (SAS: Solaris 9 or later with 9/05 update)	sparcv9 (UltraSparc)
Solaris x86	64	Sun Solaris 10 (SunOS 5.10) (SAS: Solaris 10 1/06 or later; if using Solaris 10, then apply patch 118833-27 or later)	AMD AMD64 or Intel EM64T
Win32	32	Microsoft® Windows® 2003 (NT 5.2)	Intel Pentium Pro (i686)
Win64	64	Microsoft Windows 2003 (NT 5.2)	AMD AMD64 or Intel EM64T

Linux Notes

DataFlux supports any distribution of Linux which meets the minimum requirements for kernel and glibc versions mentioned above. We do not require a specific distribution like RedHat® or SuSe. Following is a list of some of the more popular distributions and the minimum version of each which meets these requirements and is still supported by the vendor:

- Red Hat® Fedora™: 11.0
- Red Hat Enterprise Linux®: 4.0
- Novell® SuSe® Linux Enterprise Server: 9.0
- Canonical© Ubuntu©: 6.06

Supported Databases

There are numerous databases supported by the DataFlux Data Management Platform. For a list of supported databases, see the *DataFlux Data Management Studio Installation and Configuration Guide*.

Bundled UNIX Drivers

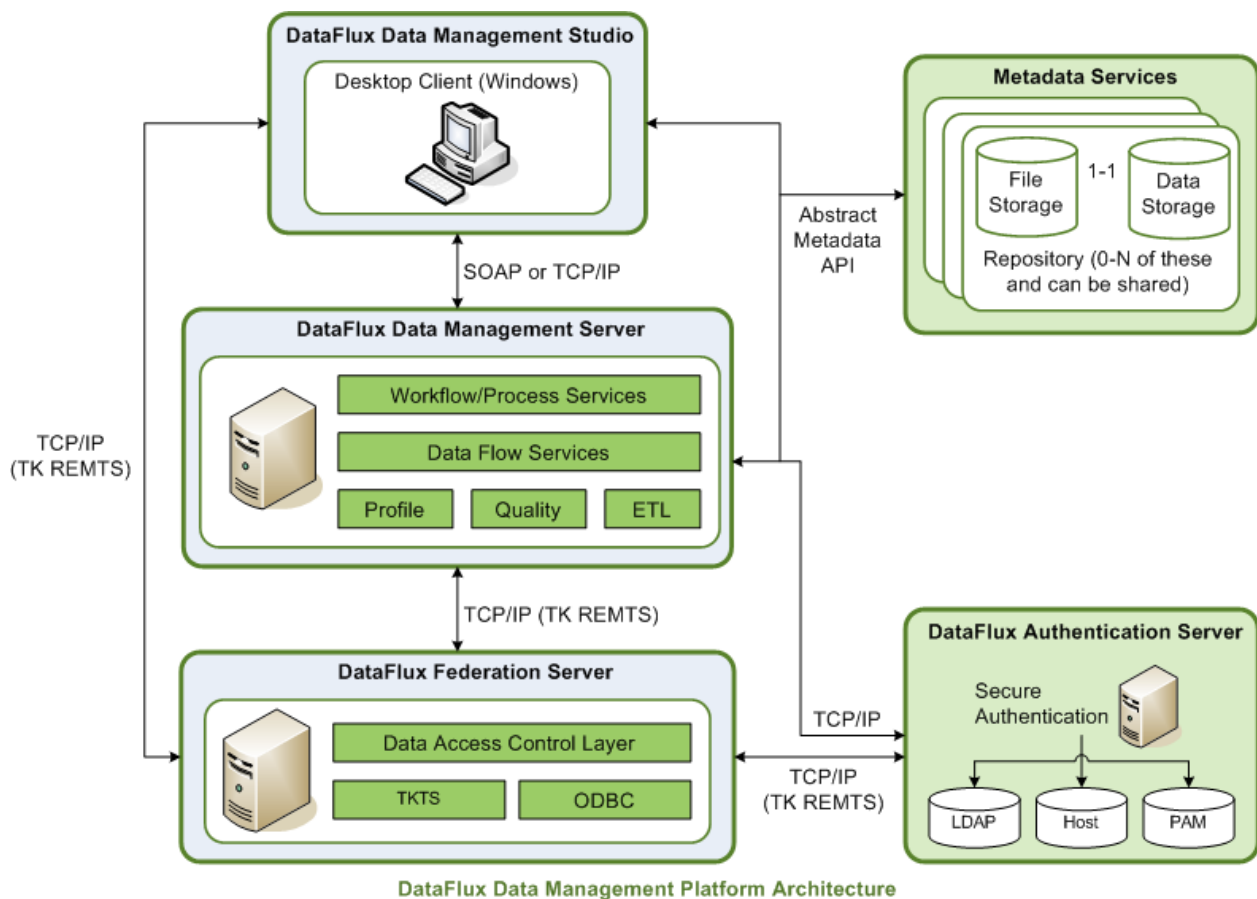
The DataFlux Data Management Platform bundles several UNIX drivers. For a list of bundled UNIX drivers, see the *DataFlux Data Management Studio Installation and Configuration Guide*.

Overview

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

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

The following figure illustrates the DataFlux Data Management Platform architecture, with the Data Management Server:



What Data Management Server Does

Data Management Server is available in two editions—Standard and Enterprise. Data Management Server – Standard supports the ability to run batch Studio jobs in a client/server environment. Data Management Server – Standard allows any Studio client user to offload batch Profile and Architect jobs to a more scalable server environment. This capability frees up resources on user's local systems. For more information, see [Data Management Server - Standard](#).

Data Management Server – Enterprise also has the capability of calling business services designed in the Studio client environment. Additionally, Enterprise invokes real-time services. For more information, see [Data Management Server - Enterprise](#).

Where and How Data Management Server Runs

Data Management Server can be deployed on Microsoft® Windows®, UNIX®, and Linux® platforms with client/server communication using HTTP. Studio users can select the **Run Job Remotely** option to have a Studio client send a job to the server.

Also included with Data Management Server is the ability to make 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. Data Management Server – Standard requires a developer to code programmatic calls to the engine.

DataFlux Data Management Server Configuration Options

Data Management Server reads configuration options from the configuration file. The installer creates the `DFEXEC_HOME\etc\dmsserver.cfg` configuration file with default values for the essential options. This file is in a "key = value" format and can be edited with any text editor.

For a complete list of available configuration settings, see [Data Management Server Configuration Directives](#).

Installing Data Management Server

This chapter describes how to install DataFlux® Data Management Server (Data Management Server) in Microsoft® Windows® and UNIX® environments.

Once you complete the installation, you need to configure the server for your environment. For more information, see [Configuring Data Management Server](#).

Download the latest version of Data Management Server for Windows or UNIX from the download section of the MyDataFlux Portal at <http://www.dataflux.com/MyDataFlux-Portal>.

Windows

To install your copy of Data Management Server using the installation wizard, complete the following steps:

1. Type the following command to start the installation program:

```
dmpversion-server-win32.exe
```

2. In the **Welcome** window, click **Next**.
3. In the **Destination Location** window, accept the default installation directory, or click **Browse** to choose a different directory. Then, click **Next**.
4. In the **Optional Components** window, select the components you want to install, and then click **Next**.
5. In the **Licensing** window, accept the default licensing method and location or select a new method and location, and then click **Next**.
6. In the **Start Installation** window, click **Next** to begin the installation.

The **Installing** window shows the progress of the installation. To cancel the installation, click **Cancel**.

7. In the **Installation Complete** window, click **Finish** to exit. If you want to view the Release Notes, select **View Release Notes**.

The default installation directory for the server is:

```
drive:\Program Files\DataFlux\DMServer\version
```

Directory Layout for Data Management Server Installation

The following table lists the directories that are created during the Data Management Server installation:

Directory	Description
Data Management Server\ <i>version</i>	Specifies the top-level installation directory.
\bin	Contains the executable files for this platform.
\data	Contains files which include data information that is specific to this installation.
\data\install	Contains a collection of .meta files, which indicate what was installed. This is useful for determining what fix packs were installed.
\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).
\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 and the ProfileExec.djf process job that is used to run the jobs on the server. If this job is missing, you will not be able to run profile jobs.
\etc\security	Contains files which specify server commands and permissions for specific users and groups.

Directory	Description
\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 Data Management Server.
\var	Contains the log files from the running of Data Management Server as well as job specific logs.
\var\repositories	Contains the sample repository file, <code>server.rps</code> .

Installing Other DataFlux Products

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

Security Considerations

Additional information coming soon.

Data Management Server Components

This chapter includes a technical description of Data Management Server and its functional components.

Data Management Server Service

DataFlux® Data Management Server (Data Management Server) runs as a Microsoft® Windows® service (called DataFlux Data Management Server). You can start and stop the service using the Microsoft Management Console (MMC). Data Management Server uses the DataFlux Data Management Studio (Studio) execution environment for Windows to execute real-time services, as well as Architect and Profile jobs.

In UNIX, Data Management Server runs as a daemon administered from a command line. The `disadmin` application is used to start and stop the daemon. Real-time services, Architect, and Profile jobs associated with Data Management Server are administered through the Studio environment.

Starting and Stopping Data Management Server in Windows


When installed in a Microsoft Windows environment, Data Management Server runs as a Windows service (named DataFlux Data Management Server).

Start and stop the service using the MMC. The MMC hosts administrative tools that you can use to administer networks, computers, services, and other system components.

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

Modifying Data Management Server Windows Service Log On

When Data Management Server is installed, it creates a service named DataFlux Data Management Server. By default, this service is started using the local system account.

 **Note:** Because this account may have some restrictions (such as accessing network drives) we suggest that you modify the service properties to have the service log on using a user account with the appropriate privileges, such as access to required network drives and files. For security reasons, you should assign administrative privileges only if necessary.

To modify the Data Management Server log on:

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

Starting and Stopping Data Management Server Daemon in UNIX

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

Command	Description
start	Allows you to start the Data Management Server. For example: <pre>./bin/disadmin start</pre>
stop	Allows you to stop the Data Management Server. For example: <pre>./bin/disadmin stop</pre>
status	Allows you to check whether the Data Management Server is running.
help	Allows you to display help information.
version	Allows you to display the version information.

Multi-threaded Operation

Data Management Server and its components operate in a multi-threaded configuration using two servers. Both servers are part of a single Data Management Server process, but run in independent threads on different ports and share the same thread pool.

The two servers are:

- **SOAP server** — Data Management Server functionality is provided via SOAP interface, as defined in the DataFlux Web Service Definition Language (WSDL). For more information on the WSDL file, see [Appendix A: Code Examples](#).
- **Wire Level Protocol (WLP) server** — Data Management Server functionality is provided via a proprietary WLP by using a proprietary WLP client library. WLP offers a significant performance increase over SOAP when communicating with the server, especially for real-time services.

For more information, see [Using Configuration Settings for Multi-threaded Operation](#).

Data Management Server – Standard

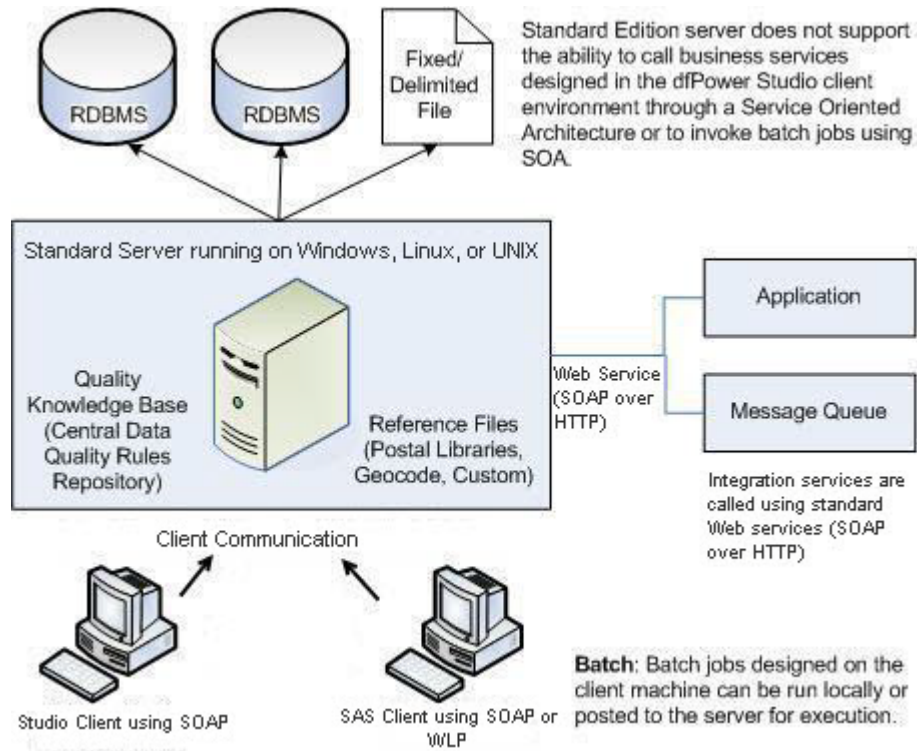
DataFlux® Data Management Server (Data Management Server)– Standard supports native programmatic interfaces for C, C++, COM, Java™, Perl, Microsoft® .NET, and Web services. The server runs in its own process as a Microsoft Windows® service or UNIX®/Linux® daemon. The Data Management Server installation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

Key Benefits of Data Management Server – Standard

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

Architecture of Data Management Server – Standard

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



Data Management Server – Enterprise

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

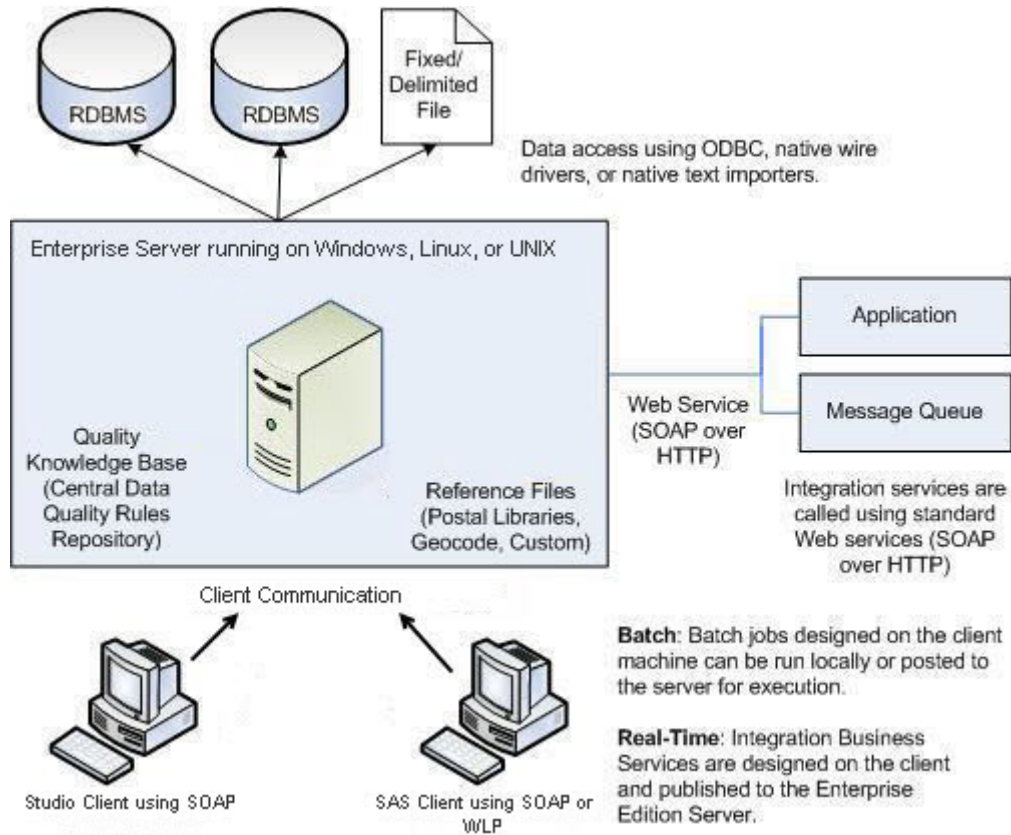
The Data Management Server – Enterprise supports real-time deployment using SOA, as well as the ability to run batch Studio jobs.

Key Benefits of Data Management Server – Enterprise

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

Architecture of Data Management Server – Enterprise

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



Understanding Data Management Server – Enterprise Processes

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

More specifically, the server handles the following processes:

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

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

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

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

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

When the server receives a request to process data or macros from a client call, it identifies an idle service, sends the data to the idle service, and listens for additional client requests. If no idle service is identified, 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.

Data Management Server Jobs

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

You can run jobs using the DataFlux Data Management Studio user interface or the command line. To run jobs from the command line, use the `dmpexec` process. For information on running jobs from the command line, see the *DataFlux Data Management Studio Online Help*.

Real-Time Service Run Types

A real-time service run occurs when you send a request to the Data Management Server to run a given service. The request may include specific data and inputs to process. Then, you wait on that same connection for a response from the server, which might include output data from the service run. If that service run takes too long to finish, the connection times out and gets dropped. In this case, you do not get anything back and you have no way of knowing what the outputs of the service run were, if there were any. You also do not know whether the service run completed successfully, such as updating a database or writing to a file.

There are two real-time service run types: data services and process services.

Real-Time Data Services

A real-time data service, which was referred to as an Architect service in previous versions of the server, uses the `dfwsvc` process to run. For information on this process, see [dfwsvc Process](#). A data service can have an External Data Provider (EDP) node, in which you can pass input data into the service to be processed. If the data service does not have an EDP node, it might receive input data from another source, depending on the nature of the service. For example, a data service can be designed to retrieve data from a database. If you have existing Architect services, you can run them as data services, without converting the old job files.

Real-Time Process Services

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

Batch Job Run Types

A batch job run occurs when you send a request to the Data Management Server to run a given batch or profile job. The job request might include inputs to process. You will immediately receive a response indicating whether or not the batch job started successfully. If the job starts, you can check the progress of the job while it is running. Once the job finishes running, check the job log file to obtain the final status of the run.

There are two batch job run types: batch jobs and profile jobs. A `dmpexec` process is used to run both batch jobs. For more information about the `dmpexec` process, see [dmpexec Process](#).

Batch Jobs

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

Profile Jobs

A profile job must reside in the Data Management Server repository. When you run a profile job, the server finds the job in the repository and starts a new `dmpexec` process to run the requested profile job. Profile jobs from previous versions of the server **cannot** be run as-is; they must be converted and imported into the new server repository. For more information on migrating profile jobs, see [Migrating Jobs, ACL Settings, and Command Permissions](#).

Data Management Server Processes

This section will describe the types of processes Data Management Server launches that run services and jobs: dfwsvc and dmpexec.

dfwsvc Process

A dfwsvc process is used to run real-time data services. The Data Management Server manages dfwsvc processes. The server knows which processes are idle and which are currently handling a user's data service request. For idle processes, the server knows what service jobs are loaded (if any). When a new request for a specific service job is received, the server first tries to find an idle dfwsvc process with that same service job already loaded. If one does not exist, the server will look for a dfwsvc process that has no job loaded. If the server does not find a dfwsvc process to reuse, a new process is started. If no additional dfwsvc processes can be started because the limit on the total number of data service processes is met, the server will try to find any idle dfwsvc process, unload the job it may have loaded, and reload a different service job into that dfwsvc process. The configuration option that sets the dfwsvc processes limit is `DMSERVER/SOAP/DATA_SVC/MAX_NUM`. For more information on this directive, see [DMSERVER/SOAP/DATA_SVC/MAX_NUM](#).

The Data Management Server also keeps track of whether the dfwsvc process was terminated, either because it crashed or was killed. If the dfwsvc process that was running a service job terminates, the Data Management Server will notify you of this problem and will log this event into server's log. If an idle dfwsvc process gets terminated, the server will log this event into its log and will be able to start a new process when another data service request is received.

dmpexec Process

A dmpexec process is used to run process services, batch jobs, and profile jobs. The processes that handle process services, and the processes that handle batch and profile job runs are managed independently, as two separate sets of processes. For process services, the Data Management Server does not manage dmpexec processes; they are managed by the pooler. The Data Management Server requests a new process from the pooler and then tells that process which service job to load and run and what the input parameters are, if any were sent in the request.

The server does not manage dmpexec processes. Therefore, it does not know which process has loaded which job. When the Data Management Server requests a new process, it receives either a newly started process or the first idle process that the pooler finds. Then, it loads a new service job into the process to run. If it does terminate, the Data Management Server will notify you of the problem.

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

Annual Licensing Notification

Data Management Server uses an annual license process to allow users to access services and run jobs. The system alerts the user when each feature's license is nearing expiration, using the following process:

1. Sixty days before a license is due to expire, a dialog will begin appearing daily in the Data Management Server Manager. It contains a list of the licensed features that are expiring as well as the number of days remaining for each feature's license. To stop the dialog from reappearing, click **Do not display this warning again**.
2. When a license reaches its expiration date, another dialog begins displaying daily, alerting the user that one or more features have expired, and that these features are now operating within a thirty-day grace period. The dialog lists the number of days left within the grace period for each feature, or if a feature has already expired and can no longer be accessed. This dialog cannot be disabled; it will continue to appear daily.
3. After the thirty-day grace period, services or jobs that are requested through Data Management Server, but have expired, no longer run.

The server log files keep records of all notification warnings generated.

Contact your DataFlux sales executive to renew your DataFlux product licenses.

Data Management Server Security

By default, DataFlux® Data Management Server (Data Management Server) runs with security disabled. In this mode, all users can make requests and run jobs and services. The DataFlux Authentication Server (Authentication Server) is not used. All Data Management Server security management requests are disabled and a SOAP fault message will be returned stating the request is disabled. Additionally, when security is disabled, the DataFlux Federation Server cannot be used and all data source names (DSNs) that are needed by jobs and services must be defined on the Data Management Server.

The following topics provide information on Data Management Server security:

- [Enabling Security](#)
- [Security Files](#)
- [Authentication](#)
- [Command Permission Bits](#)
- [Authorization](#)
- [Security Management Commands](#)
- [Requests, Command Permissions, and Access Rules](#)
- [IP-Based Security](#)

Enabling Security

An Authentication Server is required for the Data Management Server to run in secure mode. The Authentication Server controls all users and groups, and is used by the Data Management Server to authenticate users and retrieve groups to which a user is a member. In secure mode, every request to the server must contain a user name and password. Data Management Server accepts login credentials in the HTTP header of the SOAP request, using the basic HTTP authentication mechanism. The request will be rejected and a standard HTTP error will be displayed if any of the following conditions exist:

- credentials are not provided
- the Authentication Server cannot be contacted
- the Authentication Server fails to authenticate the login credentials



Note: An Advanced Encryption Standard (AES) feature can be enabled on the Authentication Server. If AES is enabled, then all clients and servers in the DataFlux Data Management Platform must apply an update. For the latest DataFlux updates, see the MyDataFlux Portal at <http://www.dataflux.com/MyDataFlux-Portal>.

Some configuration is required to run the Data Management Server in secure mode. At a minimum, the server must be configured with the Authentication Server connection information and the name of a Data Management Server administrator (admin) group. The configured Data Management Server admin group must exist on the Authentication Server before any users can log into Data Management Server. If multiple Data Management Server instances are using the same the Authentication Server, each Data Management Server can have its own admin group, or they can share the same admin group. Only one admin group can be specified per Data Management Server instance. Any Authentication Server user who is a member of that group (either directly, or via being a member of another group which in turn is a member of Data Management Server admin group) will be granted access to all commands and objects on Data Management Server, regardless of any Data Management Server permissions settings.

While the Authentication Server manages users and groups, Data Management Server manages users' access authorization. Data Management Server authorization is handled in the following steps:

Step 1: Data Management Server determines if a user has general access rights to run a specific command. For example, whether the user can list real-time data services, or upload batch jobs. If either authorization step fails, the request will be rejected and an error message will be displayed.

Step 2: This step only applies when a command operates on a specific job or service (object). For example, a user wants to run a real-time data service called "Verify_Address". If the request is not for a specific object, the user's access rights are determined by Step 1 alone. Otherwise, if the user passes Step 1, the second authorization step checks the Access Control List (ACL) for the specific object in question to determine if the user can access it.

Determining Whether Security is Enabled

To determine whether or not the Data Management Server is running with security enabled, you can send any request without including credentials. If the server is running with security enabled, an HTTP error 401 will be returned. If the server is running with security disabled, it is still possible for the request to fail, but an HTTP error will not be returned.

When the Data Management Server is running with security enabled and a request is made to get the server version, a successful response will include the Authentication Server connection string. This allows you to determine which Authentication Server instance is being used by the Data Management Server.

Security Files

Data Management Server authorization information is stored in two types of files: `users` and `groups` files and access control list (ACL) files.

The Users and Groups Files

The first type of security files are the `users` and `groups` files. Single `users` files contain specific command permissions set for individual users. Single `groups` files contain specific command permissions set for individual groups. These two files will only exist if specific command permissions are set for any users or groups, and are located in the `DFEXEC_HOME/etc/security` directory. These files are read only on Data Management Server start-up. Their contents are updated automatically when the Data Management Server Security Management (DSM) commands are used. These files should not be edited manually. If they are, the changes will be ignored and could be overwritten during Data Management Server operations.

ACL Files

The second type of security files are ACL files. Every job and service on the server has a corresponding ACL. An ACL consists of an object's owner and zero or more access control entries (ACEs). An object's owner always has access to the object, regardless of ACEs, and is allowed to change the ACL. The owner can be a user or a group. In the case of a group, every user that is a member of that group (directly or indirectly by being a member of another group which is a member of the owner-group) has access to that object and is allowed to change the ACL. Each ACE in the ACL file is a specific user or group and a value of true or false, which controls whether that user or group should have access to the object.

ACL files exist in an `.acl` directory located inside the directory that contains the object files themselves. An ACL file is read by the Data Management Server when it first accesses the corresponding object. The contents of the ACL file are updated automatically when the Data Management Server admin or object owner uses DSM commands. ACL files should not be edited manually. If they are, the changes will be ignored and may be overwritten during Data Management Server operations.

If an object exists without an ACL file, on first access to the object, a default ACL file will be created. The default ACL will have the Data Management Server admin group set as the owner and no ACEs will be defined. Therefore, the Data Management Server admin will be the only user who can then access the object and change its ACL.

Authentication

When the Data Management Server receives a request, it passes credentials from the request to the Authentication Server for authentication. Once a user is authenticated, the Data Management Server checks if the user has specific command permissions. A user has specific command permissions if one of the following conditions exist:

- The Data Management Server read the user's permissions from the `users` file on start-up
- The Data Management Server admin used DSM commands to configure permissions for that user at run-time

If a user does not have specific command permissions, the Data Management Server will apply a set of default permissions. The default permissions are controlled by the `DMSERVER/SECURE/DEF_CMD_PERMS` configuration directive. For more information on this option, see [DMSERVER/SECURE/DEF_CMD_PERMS](#). If this configuration directive is not set, all of the commands permissions will default to *inherit*. For more information on the *inherit* value, see [Command Permissions Bits](#).

PUBLIC User

In Data Management Server, a PUBLIC user is a generic category that includes all users that do not exist in the Authentication Server. The Authentication Server can authenticate a user against an Lightweight Directory Access Protocol (LDAP) server or Active Directory server, but that user might not be in the Authentication Server. PUBLIC users do not have a unique Authentication Server ID. All such users are the same to the Data Management Server and fall into the PUBLIC users category. All Data Management Server admins can set command permissions for the PUBLIC group, just as they do for any Authentication Server admin-created group.

Command Permission Bits

The following set of permissions can be applied to an individual user or group. There are three possible values for any given permission bit (1 through 16): *allow*, *deny*, and *inherit*. the *inherit* value means the specific permission is not set on the given user or group, and the permission bit's value needs to be looked up on the parent group.

DIS supports the following set of command permissions:

Bit Position	Description
1	Run a data service
2	Run a process service
3	Run a batch job
4	Run a profile job
5	Post a data service
6	Post a process service
7	Post a batch job
8	Post a profile job
9	Delete a data service
10	Delete a process service
11	Delete a batch job
12	Delete a profile job
13	List data services
14	List process services
15	List batch jobs
16	List profile jobs

For a list of which permission bit positions apply to each command, see [Command Permissions and Access Rules](#).

Authorization

Group Authorization Checks

Authorization starts when the Data Management Server checks if a user is a member of the Data Management Server administrators group. If a user is a Data Management Server admin, no further authorization checks are performed and the user gets access to all Data Management Server commands and objects, regardless of specific permission and ACL settings. If the admin group does not exist on the Authentication Server, all users will be denied access to Data Management Server. The Data Management Server requires you to have an administrators group.

Next, the Data Management Server checks if the deny group is configured. This can be set using the [DMSERVER/SECURE/GRP_DENY](#) configuration option. If it is, then if a user is a member of that group (directly or indirectly), no further authorization checks are performed and the user will be denied access to all Data Management Server commands and objects. If this group is configured on the Data Management Server but does not exist on the Authentication Server when Data Management Server is starting up, any user will be denied access to Data Management Server. For example, if the Data Management Server admin mistyped the group's name in the Data Management Server configuration file, the error will be evident, instead of being ignored. The group can be deleted from the Authentication Server after the Data Management Server has already been started; in that case, no users will be members of the deny group.

Then, the Data Management Server checks if the allow group is configured. This can be set using the [DMSERVER/SECURE/GRP_ALLOW](#) configuration option. If it is, if a user is NOT a member of that group (directly or indirectly), no further authorization checks are performed and the user will be denied access to all Data Management Server commands and objects. If this group is configured on Data Management Server but does not exist on the Authentication Server, users will be denied access to Data Management Server.

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

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

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

Group Permissions

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

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

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

If neither the user, any of groups of which the user is a member, the USERS group, nor the PUBLIC group have permission set to allow access to a given command, the Data Management Server will deny access without checking the ACL. This means the Data Management Server requires a command permission that specifically allows access to a command, before Data Management Server will look at the ACL of an individual object, if one exists.

ACL Authorization Checks

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

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

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

Security Management Commands

The Authentication Server, the admin, deny and allow groups, and default command permissions are all set in the Data Management Server configuration file, `dmserver.cfg`. The configuration file cannot be changed at run-time; any changes will be ignored until the server is restarted. Listing, setting, and deleting Data Management Server permissions for users and groups, as well as getting and setting ACLs for objects, can be done at run-time by using the Data Management Server Security Management (DSM) commands.

Access Rights to DSM

When DSM is used to set one or more of user or group permissions, the new permissions setting is committed to the `users` or `groups` file, as part of the DSM request. While setting, getting, and deleting permissions for PUBLIC group works the same way as with any other group, those requests are not supported for individual public users. The permissions for public users can only be controlled through the PUBLIC group.

An ACL for an object can be set only after an object has already been posted to Data Management Server. You cannot set an ACL as part of posting an object. When explicitly setting an ACL, the owner can be set to a user or a group. When an object is posted, the Data Management Server will automatically create a default ACL. The default ACL will have the user who posted the object set as the owner and will not contain any ACEs. The owner, or Data Management Server admin, can get and set the ACL of an object if the user has `list` command permission granted for that object type.

When an object is being posted by a public user or an ACL is being set using DSM with the owner set to a public user (the Authentication Server ID is an empty string), the Data Management Server will set the owner to Authentication Server PUBLIC group. Therefore, all users, both public and those that are not public, will be owners of that object.

When setting an ACL, Data Management Server will check for duplicate ACEs (multiple ACEs with the same Authentication Server ID) and will reject the request (will not set ACL) if duplicate ACEs are found. Also, when Data Management Server reads an ACL from a file and finds it to contain duplicate ACEs (which means the ACL file was edited manually), the ACL will be rejected and the user's access to that object will be denied. Data Management Server ACL files are to be managed via DSM commands only, and should not be edited manually.

Command Permissions and Access Rules

The following sections describe what commands specific users can run and which permission bit position apply to each command.

Command Permissions

The following table lists which users can run which of the stated commands:

User	Command
Data Management Server admin users	Set a user and group permissions
	Delete a user and group permissions
	Set a maximum number of data services that can run in parallel
	Post a repository configuration file
	Post a DSN and saved connection file
	Get a repository configuration file
	Get a DSN and saved connection file
	List a repository configuration files
	List a DSN and saved connection files
	Delete a repository configuration file
	Delete a DSN and saved connection file
Data Management Server admin user or the user who started a given batch job run instance	Get a log or statistics file for a specific batch job run instance
	Delete the history of a specific batch job run instance
	Stop a running batch job instance



Note: In order to get an ACL, the user must be allowed to list objects of that type and the ACL of the given object must grant that user access to that object. In order to set an ACL the user must be allowed to list objects of that type and the ACL must define that user as the owner.

Access Rules

The following table lists the command permission bits and the requests to which those bits apply:

Bit	Request
bit 1: execute data service	Get data service parameters
	Run a data service
	Preload a data service
	Unload a data service. If the service is busy processing data, you must be a Data Management Server admin or the owner of the data.
bit 2: execute process service	Get a data service file
	Get a process service parameters
	Run a process service
bit 3: execute batch job	Get a process service file
	Run a batch job
	Get a batch job file
bit 4: execute profile job	Get a batch job nodes' statuses
	Run a profile job
	Get a profile job file
bit 5: post data service	Upload a data service
bit 6: post process service	Upload a process service
bit 7: post batch job	Upload a batch job
bit 8: post profile job	Upload a profile job
bit 9: delete data service	Delete a data service ¹
bit 10: delete process service	Delete a process service ¹
bit 11: delete batch job	Delete a batch job ¹
bit 12: delete profile job	Delete a profile job ¹
bit 13: list data services	List data services

Bit	Request
bit 14: list process services	List process services
bit 15: list batch jobs	List batch jobs
bit 16: list profile jobs	List profile jobs

1 When deleting an object, not only must the appropriate permission bit be set to *allow*, but you must also be the owner of the object or an administrator.

IP-Based Security

You can control access to Data Management Server by IP address with configuration settings in the `dmserver.cfg` file.

There are two types of access that can be restricted by client IP address: general access and access to post and delete commands. When configuring each restriction type, you must specify either *allow* or *deny*, but not both. This directive can be followed by lists of specific IP addresses and ranges. You can also use the *all* or *none* keywords, but in this case any explicitly defined IP addresses or ranges are ignored. An IP address that is denied general access is implicitly denied access to post and delete commands.

Configuration for each restriction group must be entered on a single line using the 'space' character as a separator between entries. IP address ranges must be specified using '-' character with no spaces.

The following table lists the settings that control access, based on the client's IP address:

Setting	Description and Example
DMSERVER/IPACC/ALL_REQUESTS = (<i>allow/deny</i>)	Use this setting to restrict access to the server by IP address. If this is not set, the default is to <i>allow all</i> , which is suitable for administrators. For example: <pre>DMSERVER/IPACC/ALL_REQUESTS = allow 127.0.0.1 192.168.1.1-192.168.1.255</pre> <pre>DMSERVER/IPACC/ALL_REQUESTS = allow 127.0.0.1 192.168.1.190</pre>
DMSERVER/IPACC/POST_DELETE = (<i>allow/deny</i>)	Use this setting to control who can post and delete jobs. If this option is not set, the default is to <i>allow all</i> . For example: <pre>DMSERVER/IPACC/POST_DELETE = 127.0.0.1</pre>

Upgrade Path for Security-Related Objects

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

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

1. [Migrating Users and Groups Files](#)
2. [Migrating Jobs, ACL Settings, and Command Permissions](#)



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

Migrating Users and Groups Files

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

Migrating Jobs, ACL Settings, and Command Permissions

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

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

To migrate existing user and group command permissions, look at the `users` and `groups` files (either by opening the files in a text editor or via the DataFlux Integration Server Security Manager application) and configure the corresponding command permissions settings for the users and groups using Studio.

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

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

Configuring DataFlux Data Management Server

This section covers configuring DataFlux® Data Management Server (Data Management Server) for Microsoft® Windows® and UNIX® operating systems. Configuring the Data Management Server entails performing the following tasks:

- [Data Management Server Configuration Directives](#)
- [Configuring Licensing](#)
- [Configuring a Data Source](#)
- [Configuring the Data Access Component](#)
- [Changing Configuration Settings](#)

Data Management Server Configuration Directives

You can specify configuration settings for the DataFlux® Data Management Server (Data Management Server). These may need to be modified prior to running the server. For more information and examples, see [Using Configuration Settings](#). Additional examples of the settings can be found in the `dmserver.cfg` configuration file.

Best Practice: Refer to [Best Practices - Consider Data Management Server Performance when Modifying Configuration Settings](#) for information about performance considerations when modifying configuration settings.

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


Setting	Description
DMSERVER/THREADS/COUNT_MAX	Specifies the number of threads Data Management Server can start. The default and lowest number of threads is six. If the value is set lower than six, the value will be automatically adjusted to six. There is no setting for an unlimited number of threads. For optimal performance, the number of configured threads should be based on the expected number of parallel clients and requests.
DMSERVER/THREADS/IDLE_MAX	Specifies the number of idle threads Data Management Server can keep. The default is 0; if any thread becomes idle, it is terminated. If it is needed again, it will be restarted, which could affect the performance of the server.


Setting	Description
DMSERVER/THREADS/IDLE_TIMEOUT	Specifies the number of microseconds before a thread is flagged as idle after it stops doing work. The default is 0; threads are initially flagged as idle.
DMSERVER/SECURE	<p>Specifies whether the Data Management Server security subsystem is needed. The default is no and the secure configuration options will be ignored.</p> <p>Set the DataFlux Authentication Server (Authentication Server) connection string via the BASE/AUTH_SERVER_LOC configuration option. For more information, see the <i>DataFlux Authentication Server Administrator's Guide</i>.</p>
DMSERVER/SECURE/GRP_ADMIN	Specifies the name of the Data Management Server administrator group. If this option is defined, the group must exist on the Authentication Server. If this option is not defined or the group does not exist on the Authentication Server, secured Data Management Server will return an error during startup.
DMSERVER/SECURE/GRP_ALLOW	Specifies the name of the group whose members are allowed access to Data Management Server. This option must be defined and the group must exist on the Authentication Server. Setting this directive is optional; if no value is set, no users will be excluded.
DMSERVER/SECURE/GRP_DENY	Specifies the name of the group whose members are denied access to Data Management Server. This option must be defined and the group must exist on the Authentication Server. Setting this directive is optional; if no value is set, no users will be excluded.
DMSERVER/SECURE/DEF_CMD_PERMS	Specifies the default command authorizations to apply to users that do not have Data Management Server command authorizations explicitly set. If no value is set, by default, all authorizations will be inherited based on the group membership hierarchy, up to the PUBLIC group.

Setting	Description
DMSERVER/IPACC/ALL_REQUESTS	Controls access to all SOAP requests based on the client's IP address. By default, this option is disabled. For more information, see IP-Based Security .
DMSERVER/IPACC/POST_DELETE	Controls access to posting and deleting SOAP requests based on the client's IP address. This includes uploading new objects, such as jobs and services, to Data Management Server and deleting existing objects from the server. By default, this option is disabled. For more information, see IP-Based Security .
DMSERVER/JOBS_ROOT_PATH	Specifies the root directory under which directories with jobs and services are located. The default object root directory is DMSERVER/WORK_ROOT_PATH . If this directory does not exist, it will be created. The directories with jobs and services are: Data Services, Process Services, and Batch Jobs. If these do not exist, they will also be created.
DMSERVER/PROC_JOB/MAX_NUM	Specifies the maximum number of batch jobs that Data Management Server can run simultaneously. The default is 10. If a new job request is issued that will exceed the limit, an error message will be displayed.
DMSERVER/DATA_SVC/DEBUG	Specifies whether the dfwsvc and dmpexec real-time service processes will generate a log file. If this option is set to yes, dfwsvc will log debug-level messages, while dmpexec will log messages at the level that is set in the <code>dfwfproc.log.xml</code> file. For performance reasons, the default value is no.
DMSERVER/DATA_SVC/LOG	Specifies whether the dfwsvc and dmpexec real-time service processes will generate a log file. If this option is set to yes, dfwsvc will log debug-level messages, while dmpexec will log messages at the level that is set in the <code>dfwfproc.log.xml</code> file. For performance reasons, the default value is no.
DMSERVER/LOG_CHUNK_SIZE	Controls the size of each log file or statistics file chunk that is sent back to the client from the getJobLog request. For log file, this option controls the number of characters per chunk. For statistics files, this option controls the number of bytes per chunk. The default value is 512K.

Setting	Description
DMSERVER/WORK_ROOT_PATH	<p>Specifies the root directory under which Data Management Server work directories will be created. Each time the server starts, a new work directory will be created for that instance of the server. The name of this directory contains the server startup date and time, as well as the corresponding process ID. The Data Management Server work directory contains the following files:</p> <ul style="list-style-type: none"> • Data Management Server log files • dfwsvc log files • dmpexec log files • statistics files • shared memory files for Data Management Server and dfwsvc communication • temporary files that are generated by running jobs and services. <p>The default directory is DFEXEC_HOME/var/.</p>
DMSERVER/SOAP/LISTEN_PORT	<p>Specifies the port on which the SOAP server will listen for connections. The default port is 21036.</p>
DMSERVER/SOAP/CONNS_BACKLOG	<p>Specifies the maximum size of the connection request queue. The backlog is used when the SOAP server receives more connections than it can accept and process within a specific timeframe. The default is 100.</p>
DMSERVER/SOAP/RDWR_TIMEOUT	<p>Specifies the timeout value for read and write operations on a socket. Set a positive value to seconds, a negative value to microseconds, and no timeout to 0. If a non-0 value is set, a timeout will occur if no data can be sent or received within the configured time after the server initiates a send or receive operation over the socket. When a timeout occurs, a SOAP_EOF error will be returned. The default is 0.5 seconds.</p>

Setting	Description
DMSERVER/SOAP/RETURN_NULLS	Specifies what the real-time data service will return for output fields. If set to yes, NULLs will be returned; if set to no, empty strings will be returned. The default value is no.
DMSERVER/SOAP/LOG_PACKETS	Specifies whether to generate the _PACKETS_ log file. If set to yes, the log file will be generated in the same directory as the input, output, and error SOAP packets log files. For performance reasons, the default value is no.
DMSERVER/SOAP/SSL	<p>Specifies whether to enable SOAP communication over SSL. The default value is no. The following SSL configuration directives come into play when SSL is enabled:</p> <ul style="list-style-type: none"> • DMSERVER/SOAP/SSL/KEY_FILE • DMSERVER/SOAP/SSL/KEY_PASSWD • DMSERVER/SOAP/SSL/CA_CERT_FILE • DMSERVER/SOAP/SSL/CA_CERT_PATH
DMSERVER/SOAP/SSL/KEY_FILE	Specifies the path to the key file that is required when the SOAP server must authenticate to clients. If this configuration directive is not used, comment it out.
DMSERVER/SOAP/SSL/KEY_PASSWD	Specifies the password for DMSERVER/SOAP/SSL/KEY_FILE . If the key file is not password protected, this configuration directive should be commented out.
DMSERVER/SOAP/SSL/CA_CERT_FILE	Specifies the file where the Certificates Authority stores trusted certificates. If this configuration directives is not needed, comment it out.
DMSERVER/SOAP/SSL/CA_CERT_PATH	Specifies the path to the directory where trusted certificates are stored. If this configuration directive is not needed, comment it out.

Setting	Description
DMSERVER/SOAP/DATA_SVC/QUEUE	<p>Specifies whether to queue real-time data service requests. If set to yes and all running dfwsvc processes are busy and Data Management Server is not allowed to start another one to handle a new service request, the request will be put in a queue. As soon as a dfwsvc process becomes available, the request will be handled. The default is no and in the above scenario, an error message will be displayed.</p>
DMSERVER/SOAP/DATA_SVC/MAX_NUM	<p>Specifies the maximum number of real-time data services the SOAP server will allow to run simultaneously. The default is ten. If a new service request would exceed the set limit and queue is not enabled, an error message will be displayed.</p> <p> NOTE: This option applies to the SOAP server, meaning the service requests are coming from a SOAP client. It does not apply to the WLP server or requests coming from a WLP client.</p>
DMSERVER/SOAP/DATA_SVC/MAX_ERRS	<p>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 this is no limit.</p>
DMSERVER/SOAP/DATA_SVC/MAX_REQUESTS	<p>Specifies the maximum number of service requests a dfwsvc process is allowed to handle before it is forced to terminate. The default is -1, meaning this is no limit.</p>
DMSERVER/SOAP/DATA_SVC/PRELOAD_ALL	<p>Specifies that the Data Management Server should find and preload all services a specific number of times. This includes services found in subdirectories. The number of instances specified must be an integer greater than zero, or the directive is ignored. This can be used with dfsvc preload. For more information, see Pre-loading Services.</p>

Setting	Description
DMSERVER/SOAP/DATA_SVC/PRELOAD	Specifies services and the count for each service that Data Management Server pre-loads during startup. This can be used with dfsvc preload all . For more information and formatting guidelines, see Pre-loading Services .
DMSERVER/SOAP/PROC_SVC/MAX_NUM	Specifies the maximum number of real-time process services that Data Management Server will allow to run simultaneously. The default is ten. If a new service request will exceed this limit, an error message will be displayed.
DMSERVER/WLP/DATA_SVC/MAX_NUM	<p>Specifies the maximum number of real-time data services the WLP server will allow to run simultaneously. The default is ten. If a new service request would exceed this limit, an error message will be displayed.</p> <p> NOTE: : This option applies to the WLP server, meaning the service requests are coming from a WLP client. It does not apply to the SOAP server or requests coming from a SOAP client.</p>
DMSERVER/WLP/CHILD/LISTEN	Specifies the communication method between dfwsvc and Data Management Server. This option takes precedence over the DMSERVER/WLP/CHILD/SHM_DIR and DMSERVER/WLP/CHILD/TCP_PORT options. If you do not configure this option, it will default to the shared memory provider and the shared memory files will be created in the Data Management Server work directory, DFEXEC_HOME/var/.
DMSERVER/WLP/CHILD/TCP_PORT	Specifies the port number for TCP on the host machine. This is used when DMSERVER/WLP/CHILD/LISTEN is not set and the default method is TCP. If the default method is not TCP, do not specify this option, as it will select the TCP method. The default port is 21035.


Setting	Description
DMSERVER/WLP/CHILD/SHM_DIR	Specifies the directory in which to create SHM files. This is used when DMSERVER/WLP/CHILD/LISTEN is not set and the default method is SHM. If the default method is not SHM, do not specify this option, as it will select the SHM method. This option takes precedence over the DMSERVER/WLP/CHILD/TCP_PORT option. The default directory is the Data Management Server work directory, DFEXEC_HOME/var/.
DMSERVER/WLP/SVR/LISTEN	Specifies the connection information that clients will use to contact the WLP listen server when WLP Data Management Server client or SAS WLP Batch client access is required.
DMSERVER/WLP/SVR/LISTEN_PORT	Specifies the port number for TCP on the host machine. This is used when DMSERVER/WLP/SVR/LISTEN is not set. The default port is 21037.

Configuring Licensing

Data Management Server uses a file-based licensing model that takes the form of a machine-specific license file. The license pool for executing jobs and services using Data Management Server has uncounted licenses (an infinite number of licenses) for each type of license purchased. If Data Management Server is packaged as part of SAS, you have the option of selecting SAS license file as your licensing method.

Windows

To configure your license file for Data Management Server in Windows, complete the following steps:

 **Note:** The license dir parameter in the `dfexec.cfg` file is no longer supported on Data Management Server for Microsoft Windows. In order to set or change the license location, you must use the license manager application.

1. Run the DataFlux Host ID application to generate a Host ID for your Data Management Server. From the DataFlux Data Management Studio main menu, click **Help > DataFlux Host ID**.
2. Contact your DataFlux representative and provide the DataFlux Host ID to obtain your license file.

3. Save the license file to *install_drive*: \Program Files\DataFlux\Data Management Server\version\etc\license_file.
4. Make note of the full path to the licensing location, including the file name. To specify the licensing location by using the License Manager, click **Start > Programs > DataFlux Data Management Server > License Manager**. In the **License Manager** dialog, select **DataFlux license file**, and enter the **Location**.

UNIX

To configure your license file for Data Management Server, complete the following steps:

1. To generate a Host ID, run the following command:

```
./bin/lmhostid
```

Write down the FLEXnet host ID that is returned.

2. Log onto the MyDataFlux Portal at <http://www.dataflux.com/Customer-Care> and click **Request License Unlock Codes**. This opens the License Request Form page.
3. Enter the requested information, including the Host ID generated in Step 1.
4. When you receive your new license file, save it on the UNIX® server in the *etc/license* directory. License files must have a *.lic* file name extension in order to be considered.

With file-based licensing, you should not change the license location setting in the *dfexec.cfg* configuration file.

If you need to change the licensing method, run the following command:

```
./bin/dflm
```

The optional **-m** switch allows you to change licensing methods. If you use this switch, you must restart Data Management Server.

SAS License

If you have obtained a license from SAS, complete these steps:

1. Set the license location setting in the *dfexec.cfg* configuration file to point to your license file.
2. Run the following command:

```
./bin/dflm -m
```

3. Set the license type to **SAS license file**.

Configuring a Data Source

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

For information on configuring ODBC, Federated Server, and SAS Data Set connections, see the *DataFlux Data Management Studio Installation and Configuration Guide*.

Configuring Saved Connections

After you configure the data sources and test the connection, you should store a saved connection for that data source. The DataFlux Connection Manager allows you to save connection information. Saved connections provide a mechanism for storing encrypted authentication information for a data source. When a saved connection is used, only the DSN is stored in the job file, not the entire connection string. When you run the job, it refers to the connection file for the login credentials for that DSN. In order to use a saved connection, the same connection must also be saved on the client machine where the job was created. For more information on the Connection Manager, see *DataFlux Data Management Server User's Guide*.

Best Practice: Refer to [Appendix A: Best Practices - Use Connection Manager to Configure Data Sources](#) for additional information about Configuration Settings.

Configuring the Data Access Component

The Data Access Component (DAC) allows DataFlux® Data Management Server (Data Management Server) to communicate with databases and manipulate data, by connecting to sources using Open Database Connectivity (ODBC) and Threaded Kernel Table Services (TKTS). ODBC database source names (DSNs) are not managed by the DAC, but by the Microsoft ODBC Administrator. TKTS DSNs, however, are managed by the DAC and TKTS connections are stored in a TKTS DSN directory.

DAC Configuration Options

Most of the DAC settings come from configuration values which are specified in the configuration files, including the `app.cfg` and `macro.cfg` files. For a comprehensive list of configuration values, see the *Data Management Studio Online Help*.

Changing Configuration Settings

Once you have completed the installation process, modify the `dfwsvc.cfg` file to run real-time data services and the `dfwfproc.cfg` file to run batch jobs. Set the directory paths for any relevant reference data, for example, United States Postal Service (USPS), Canada Post, Geocoding, and Quality Knowledge Base (QKB). You can also change the default port on which the server is listening. Other settings in these files control memory allocation and enhance clustering performance.

Modifying Default Configuration Settings

After installing Data Management Server, you may have to modify some default configuration settings in order for the jobs and services to run correctly. The `dfwsvc.cfg` file contains configuration settings for real-time services, as well as Architect and Profile jobs. This file is stored in the `\etc` directory of the Data Management Server installation.

Refer to [Data Management Server Configuration Directives](#) for a list of common settings that may need to be modified before running the server. After making changes to the configuration file, you must restart the server. For more information, see the [Data Management Server Service](#).



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

Defining Macros

The `macros.cfg` configuration file defines macro values for substitution into Architect jobs, and overrides predefined values. This file is located in the `etc` directory of the Data Management Server installation. Each line represents a macro value in the form `KEY = VALUE`, where the `KEY` is the macro name and `VALUE` is its value. For example, on a Windows system:

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

On a UNIX system:

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

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



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

For more information on macros, see the *DataFlux Data Management Studio Online Help*.

SOAP Commands and WSDL Types

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

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

Command	Description	Command Type
ServerVersion	Allows you to retrieve the server's version information, as well as versions of installed reference data, the repository, and some of the libraries.	Single version command
ArchitectServiceParam	Allows you to retrieve input and output fields of a data service.	Data services command
ArchitectService	Allows you to run a data service.	Data services command
ArchitectServicePreload	Allows you to start the requested number of data service processes and load specified data service jobs into those processes.	Data services command
ArchitectServiceUnload	Allows you to terminate the specified data service process.	Data services command
LoadedObjectList	Allows you to retrieve a list of running data service processes, along with the name of the loaded service job for each process.	Data services command
MaxNumJobs	Allows you to set the maximum number of concurrent data service processes. This is a run-time setting only and has no effect on the value in <code>dmserver.cfg</code> file.	Data services command
WorkFlowJobParams	Allows you to retrieve inputs and outputs of either a process service or a batch job.	Process services command

Command	Description	Command Type
WorkFlowService	Allows you to run a process service.	Process services command
RunArchitectJob	Allows you to run a batch job.	Batch and profile jobs commands
RunProfileJob	Allows you to run a profile job.	Batch and profile jobs commands
TerminateJob	Allows you to terminate a running batch or profile job. The client can still get the status, log, and statistics file (if one exists) after the job has been terminated.	Batch and profile jobs commands
JobStatus	Allows you to retrieve status information for one or more batch or profile jobs. Applies jobs that are running or already finished.	Batch and profile jobs commands
JobNodesStatus	Allows you to retrieve status information for every node in a batch job. Applies only to the jobs that are currently running.	Batch and profile jobs commands
JobLog	Allows you to retrieve the log file and statistics file (if one exists) for a batch or profile job. Applies only to already finished jobs.	Batch and profile jobs commands
DeleteJobLog	Allows you to delete a job log and statistics file (if one exists). This command removes all history for a given job run. Applies only to already finished jobs.	Batch and profile jobs commands
ObjectList	Allows you to retrieve a list of available objects of a specified type (data services, process services, batch jobs, or profile jobs).	Object files commands
PostObject	Allows you to upload an object of a specified type. If an object of that type with the same name and path already exists, an error is returned.	Object files commands
ObjFile	Allows you to download an object of a specified type.	Object files commands

Command	Description	Command Type
DeleteObject	Allows you to delete an existing object of a particular type from the server.	Object files commands
ListAccounts	Allows you to retrieve a list of user and group IDs with explicitly configured server commands permissions (which are included).	Security commands
SetPermissions	Allows you to set server command permissions for an user or group ID.	Security commands
DeleteAccount	Allows you to delete server command permissions for an user or group ID.	Security commands
SetACL	Allows you to set an access control list (ACL) for an object (data service, process service, or batch job).	Security commands
GetACL	Allows you to retrieve an ACL for an object.	Security commands

Technical Support

This section addresses questions and issues related to DataFlux® Data Management Server (Data Management Server):

[Best Practices](#)

[Frequently Asked Questions](#)

[Troubleshooting](#)

[Error Messages](#)

If you do not find your answer, please contact [DataFlux Technical Support](#).

Best Practices

Use a System Data Source Rather Than a User Data Source

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

Use Connection Manager to Configure Data Sources

When developing DataFlux® Data Management Studio (Studio) Architect jobs or services, use the Connection Manager to set up and store connection information to any Open Database Connectivity (ODBC) data source.

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

In this way, the connection information to the master reference database is made permanent and independent of the Architect job or service. The saved information does not have to be entered each time the job is run, and that information can be used by any DataFlux application.

Windows — This connection information is stored in the directory specified in one or both of the following registry keys:

- HKEY_CURRENT_USER\Software\DataFlux\dac*version**saved_connection_dir*
- HKEY_LOCAL_MACHINE\Software\DataFlux\dac*version**saved_connection_dir*

where *version* indicates the version of Data Management Server that you have installed and *saved_connection_dir* is the directory where the connection is saved.

If neither of these keys exist, connection strings are saved in the *\version* folder on the job's run-time machine.

UNIX — The connection information is saved to a file in the `/$HOME/.dfpower/dsn` directory.

Plan Your Security Model Based on Business Needs

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

For more information, see [Security Policy Planning](#).

Consider Data Management Server Performance when Modifying Configuration Settings

Changes to several of the configuration settings in Data Management Server can affect performance. For example, large temporary files, log files, and memory clusters can slow down the server. Several settings that are in the `dfexec.cfg` file or can be added to `dfexec.cfg` can alter memory allocation or processing power.

For more information, see [Data Management Server Configuration Directives](#).

Frequently Asked Questions

Need to include information on where to look when DMS win32 does not start.

The following topics provide answers to frequently asked questions:

- [General](#)
- [Installation](#)
- [Security](#)

General

What is a Data Management Server?

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

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

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

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

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

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

How can I run multiple versions of Data Management Server in Windows or UNIX?

The following procedure shows how to run multiple versions of Data Management Server on a UNIX machine. This procedure can also be applied to versions of Data Management Server older than 8.0.

UNIX

Multiple versions or instances of the same version of Data Management Server can be installed on a UNIX server. Different versions or instances must be installed in separate directories. Instead of creating a single directory for the software, for example, `/opt/dataflux`, each version or instance must have a separate directory, for example, `/opt/dataflux/dmserverv2.1`, `/opt/dataflux/dmserverv2.2`, and `/opt/dataflux/dmserverv2.3`. The installer must run in each directory and a different port must be designated for each server, otherwise the installations can be configured identically.

WINDOWS

Multiple versions of Data Management Server services can be installed on Windows systems with some modifications. When installing any version of a DataFlux Windows service, the settings of the currently installed version are overwritten, because the same names are used, preventing multiple versions from existing concurrently. The solution is to rename older versions of the services, so that installing or reinstalling the latest version does not affect the older versions.



Note: Once an older version of a service is renamed, reinstalling that version or applying an update to the version will require some user intervention, such as stopping and starting the service manually, and editing the registry so the current version of the service is pointing to the correct executable file.

The following procedure shows how to run both the 8.0 and 8.1 versions of the Data Management Server service on a Windows machine.



Note: This procedure can also be applied to versions of Data Management Server older than 8.0.

Make sure the older (8.0) Data Management Server service is the active service

1. To open the Services management console, click **Start > Settings > Control Panel**.
2. Double-click **Administrative Tools > Services**.
3. Double-click the **DataFlux Integration Server** service entry to display the **Properties** dialog.
4. Notice the Path to executable property. If that property uses the 8.0 bin directory, then skip to *Rename the older (8.0) Batch Scheduler service*, otherwise continue with this procedure. The remainder of this procedure assumes the Path to executable property uses the 8.1 bin directory. If not, substitute the appropriate directory name.
5. To stop the Data Management Server service, click **Stop**, or select **Action > Stop** from the menu.
6. Close the **Properties** page.
7. Exit the **Services** management console.
8. Open a command prompt window and change the directory to the Data Management Server 8.1 bin directory.
9. Type **dfintgsvr.exe -u** and press **Enter**. A message will appear, confirming the removal of the service.
10. Change to the Data Management Server 8.0 bin directory.
11. Type **dfintgsvr.exe -i** and press **Enter**. A message will appear, confirming the installation of the service.
12. Close the command prompt window.

Rename the older (8.0) Batch Scheduler service

1. To open the Services management console, click **Start > Settings > Control Panel**.
2. Double-click **Administrative Tools > Services**.
3. If the DataFlux Integration Server service, **DFIntegrationServiceservice**, is started, right-click on it and select **Stop**. Services can also be started and stopped by opening a command prompt window and typing **NET START service_name** or **NET STOP service_name**.
4. Exit the **Services** management console.
5. To open the registry editor, click **Start > Run**, type **regedit**, and press **Enter**.

6. Right-click on HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DFIntegrationService and select **Rename**.
7. Change the key name from DFIntegrationService to **DFIntegrationService80**.
8. Double-click the DisplayName property, listed on the right.
9. Change the name from DataFlux Integration Server to **DataFlux Integration Server 8.0**.
10. Close the registry editor.
11. Reboot the computer.



Note: Windows XP and Windows Vista® have a Services Control Manager (SCM) that manages all services running on the system. The SCM cannot be easily reset or restarted while Windows is running, making a system reboot necessary.

Reinstall the latest (8.1) Batch Scheduler service

1. Open a command prompt window and change the directory to the Data Management Server bin directory.
2. Type **dfintgsvr.exe -i** and press **Enter**. A message confirming installation of the service will appear.
3. Close the command prompt window.
4. Open the **Services** management console.
5. Right-click on the Data Management Server service and select **Start**. Both Data Management Server services, DataFlux Data Management Server and DataFlux Data Management Server 8.0, should now be in the **Services** management console with a status of Started.
6. Exit the **Services** management console.

How do I move an Architect or Profile job to UNIX so it can be processed by a Data Management Server?

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

What SOAP commands are recognized by Data Management Server?

For a complete list of SOAP commands recognized by Data Management Server, see [Appendix C: SOAP Commands](#).

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

Data Management Server is compatible with most ODBC compliant data sources. DataFlux recommends using supplied ODBC drivers instead of client-specific drivers provided by the manufacturer. Limited support will be available for implementation and problem resolution when a client implements a driver not supplied by DataFlux.

For a complete list of supported drivers, see [Supported Databases](#).

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

In Windows, a job that will be run through Data Management Server must be saved in a location that does not use mapped drives. Win32 service is not able to access mapped drives, even if the service is started under a user account that has those drives mapped.

Are there restrictions on which characters are allowed in job names?

Job names can include alphanumeric characters only. If any of the following characters are included in a job name, Data Management Server will not list that job name and will not allow any operations on that file:

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

How can I be automatically notified of new releases of DataFlux products?

New product releases, patches, and data updates are announced via customer communications and on the MyDataFlux Portal at: <http://www.dataflux.com/MyDataFlux-Portal>.

How do I know which log file is mine?

All Data Management Server log file names start with the date that corresponds to when the current instance of Data Management Server was started. The beginning of the file name will be in the following format: YYYYMMDD-HH.MM_. The date portion of the file name is followed by "00_" for the purpose of sorting the directory so that all log files related to a particular instance of Data Management Server are grouped together. The Data Management Server log itself will either be the first or last log file. After that there are some random characters the operating system creates to guarantee unique file names. The name ends with "_dmserver.log."

Service and job log files begin with: MMDD-HH.MM.SS_ representing the date when the request to load this job or service was received by Data Management Server. This is followed by a request number (since multiple requests can be received in the same second), followed by the job or service name.

For example:

```
20070215-12.13_00_0210B0_dmserver.log
20070215-12.13_0215-12.13.44_4_archsvc_ram.dmc.log
20070215-12.13_0215-12.14.04_10_archjob_100Kc1ust.dmc.log
```

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

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

To execute the command directly, type:

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

or to execute from the UNIX shell, type:

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

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

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

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

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

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

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

Setting	Default Value
DAC/DSN	<code>drive:\Program Files\DataFlux\Data Management Server\etc\dftkdsn\</code>
DAC/SAVEDCONNSYSTEM	<code>drive:\Program Files\DataFlux\Data Management Server\etc\dsn\</code>

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

How do I point the Data Management Server to a repository?

The repository configuration file (`.rcf` file) contains repository definition information. To create a custom repository definition file for the Data Management Server to point to, use the Studio user interface to create the new definition. Then, manually copy the definition file from the Studio repositories directory (`drive:\Program Files\DataFlux\DMStudio\version\etc\repositories`) to the Data Management Server repositories directory (`drive:\Program Files\DataFlux\DMServer\version\etc\repositories`). For more information on repositories and creating the repository configuration file, see the *DataFlux Data Management Studio Online Help*.



Note: The Data Management Server can only point to one repository at a time.

Installation

What is the default temp directory, and how do I change it?

Data Management Server uses the system temporary directory which is determined by the TMP environment variable in Windows and the TMPDIR environment variable in UNIX.

In UNIX, you can redirect temporary files, by doing the following:

1. Export `TMPDIR=your_new_temp_path`.
2. Edit the `.profile` file to include this new path.
3. Restart the Data Management Server daemon.

In Windows, the default setting for TEMP is `drive:\Windows\Temp`. You may set the value of TEMP Environment Variable for that user or set the TEMP System variable to a different location. To change environment variables in Windows, do the following:

1. Right-click **My Computer** and select **Properties**.
2. Click the **Advanced** tab.
3. Click **Environment variables**.
4. Click one of the following options, for either a user or a system variable:
 - Click **New** to add a new variable name and value.
 - Click an existing variable, and then click **Edit** to change its name or value.
 - Click an existing variable, and then click **Delete** to remove it.
5. Restart the Data Management Server service.

How do I connect to a database?

Data Management Server connects to databases through ODBC. To add a data source, use the ODBC Data Source Administrator provided with Windows, or use the `dfdbconf` command in UNIX. In Windows, click **Start** > **Programs** > **DataFlux Data Management Studio version** > **DataDirect Connect ODBC Help**. You may require assistance from your network administrator to install this ODBC Connection, as it requires site-specific information. For more information, see [Configuring a Data Source](#).

Security

What are the requirements for user names?

A user name is case sensitive, can be up to 20 characters, and can only include alphanumeric characters and these symbols: `.`, `-`, `_`.

Are there any restrictions for passwords?

There are no restrictions on the characters or words that can be used for passwords. A password may be set to blank. Passwords do not expire. This is also true for passwords for which you create password hashes using **HashPassword.exe**. If you are using **HashPasswordStrong.exe** to generate password hashes, your password will need to contain a minimum of: six characters, one numeric digit, one uppercase letter, and one lowercase letter.

We do not need additional security, is it required?

No, you are not required to use the Data Management Server security subsystem. It is available to fit your business needs. Security is disabled by default.

Is there a limit to the number of users and groups I can add using Data Management Server security?

There is no limit when using the Data Management Server security command line option to add users and groups however, if you are using Data Management Server Security Manager to create users and groups, there is a limit of 10,000 users and 10,000 groups.

Troubleshooting

I saved a job in DataFlux® Data Management Studio, but now I don't see it.

Check the job name. Job names can include alphanumeric characters only. If any of the following characters are included in a job name, DataFlux® Data Management Server (Data Management Server) will not list that job name and will not allow any operations on that file: , . ' [] { } () + = _ - ^ % \$ @ or !.

I can connect to the machine running the license server, but I cannot get a license.

Typically, no license server port number needs to be explicitly specified. The license server automatically selects an available port between 27000 and 27009. If a client machine can connect to the license server, but the license client process cannot connect to the license server process, an explicit port number must be specified.

In the license file on the server, a port number can be explicitly specified as the last item in the SERVER line as shown below. The port number can be added or changed in the existing license file without the need to regenerate it. The license server has to be restarted once a port number is added or changed. For example:

```
SERVER server_name.com 001125c43cba 27000
```

On the client side, a port number can be explicitly specified by prepending it to the server name, for example:

```
27000@server_name.com
```

I get one or both of the following error messages:

*[time of error] (DATAFLUX) UNSUPPORTED: "DFLLDIAG2" (PORT_AT_HOST_PLUS)
phamj4@UTIL0H4GHXD1 (License server system does not support this feature. (-
18,327))*

*[time of error] (DATAFLUX) UNSUPPORTED: "DFLLDIAG1" (PORT_AT_HOST_PLUS)
phamj4@UTIL0H4GHXD1 (License server system does not support this feature. (-
18,327))*

These error messages refer to the licenses for two built-in features that are used internally by DataFlux for debugging. These licenses are not distributed, so the license checkout request process for these two licenses fails and produces the errors noted. This is normal and should occur only once, when the license checkout request is made for the first time.

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

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

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

`server send log chunk size = 32KB`

What do the dfexec return codes mean:

When `dfexec` is called from an external program, such as a scheduler, it produces return codes to indicate its status. The return codes and their descriptions are listed here:

Return Code	Description
0	Specifies the job is still running.
1	Specifies the job has finished successfully.
2	Specifies the job has finished with errors: Unspecified internal error.
3	Specifies the job has finished with errors: Invalid command-line parameters.
4	Specifies the job has finished with errors: Invalid configuration.
5	Specifies the job has finished with errors: Failed during job execution.

Return Code	Description
6	Specifies the job has finished with errors: Licensing error.
7	Specifies the job has finished with errors: Invalid or unsupported locale.
8	Specifies the job has crashed.
9	Specifies the job was terminated.

Error Messages

The following topics specify information about Data Management Server error messages:

[Installation and Configuration](#)

[Security](#)

[Running Jobs and Real-Time Services](#)

Installation and Configuration

No Valid Base License Found

If you do not have a valid license file on your machine, you will get an error when attempting to run a job with nodes that require that license. Following is an example of an error you might expect in this case:

```
dtengine :: warning :: No valid base license found
dtengine :: error :: Job contains node with unknown id 'SAMPLEFIELDNAME'. The
node may not be licensed, or the plugin for the node may be unavailable.
dfexec :: error :: unable to load job
dfexec :: error :: aborted due to errors
```

For more information on configuring licensing, see [Configuring Licensing](#).

Data Source Connection Errors

When configuring a new data source, it is critical that parameters (such as DSN, host, port, and sid) match exactly those used to create the job on the client machine. If the connection fails, DataFlux® Data Management Server (Data Management Server) displays error messages describing the reasons for the failure.

The following table lists the common World Address verification error codes:

Error Code	Description
-100	Time for testing is expired.
156	Too many results after validation. Only the first 20 results will be presented.
157	No certification for this country.
204	Country not recognized.
205	Country database not found.
206	Country database in the wrong format or data corrupt.
207	Country database access denied. License may be missing.
300	No country rule could be loaded.
-1004	Country is locked.
-9999	Call encountered an error. The reason for the error is unknown.

Security

401 Unauthorized

If the user is not authenticated, there will be an HTTP error, 401 Unauthorized. This could mean that you have entered invalid user name and password credentials, or your user account has not been set up. Contact your Data Management Server Security Administrator for assistance.

403 Forbidden

When a user receives the HTTP error, 403 Forbidden, you have entered the system but do not have authorizations to execute a particular Data Management Server command. Contact your administrator for assistance.

Logging

An owner of an object may view the log and status for that object, but users must have authorizations to view other user logs.

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:

```
dtengine :: error :: Blue Fusion load scheme 'CUSTOM_SCHEME_1.sch' failed: Blue
Fusion error -801: file i/o error
dfexec :: error :: unable to initialise step: Standardization 2
dfexec :: error :: aborted due to errors
```

You must ensure that: (1) the Quality Knowledge Base (QKB) you are using on the Data Management Server is an exact copy of the QKB used on Studio, and (2) the name of the scheme is typed correctly, as it is case sensitive. To copy the QKB from Microsoft® Windows® to UNIX®, use FTP or Samba mappings. You must restart the Data Management Server service and retry the job. For more information, see [Data Management Server Service](#). On some UNIX systems, there is a case sensitivity issue with the schemes. Once you copy the QKB over to the UNIX server, make sure that the name of the scheme is modified to all lowercase. It is located in the qkb directory, under /scheme.

ActiveX Control Required to View Help Files

In Microsoft Internet Explorer® 6.0 and later, your network administrator can block ActiveX® controls from being downloaded. Security for ActiveX content from CDs and local files can be changed under Internet Options.

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

Locale Not Licensed

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

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

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

Node Not Licensed

An error message similar to the following can occur when the user has more than one copy of the license file:

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

Check the \license directory of the Studio installation. Remove any extra copies of studio.lic file from that folder.

Running Jobs and Real-Time Services

Error When Connecting to Excel Database

Because Microsoft Excel® is not a true database, you may occasionally experience problems making the ODBC connection. This will produce an error similar to the following:

```
Failure while connecting to data source (Excel files) : [HY000][Microsoft][ODBC Excel Driver] Cannot open database '(unknown)'. It may not be a database that your application recognizes, or the file may be corrupt. (=1028)
```

Naming the spreadsheet can fix this issue. To eliminate the error, highlight all of the cells in the spreadsheet. Select **Insert > Name** and enter the name of the spreadsheet.

Functionality Disabled Error When Creating a CASS PS Form 3553

Why did I get a "functionality disabled" error message when creating a CASS PS Form 3553?

The PS Form 3553 for CASS certification can be created only on Microsoft® Windows®, Solaris™, and AIX™ systems.

Low Disk Space on UNIX

The following error indicates that temporary disk space on the machine running Data Management Server is low:

```
dfexec :: error :: unable to read row 0
dtengine :: error :: File I/O operation failed. Possibly out of disk space.
dfexec :: error :: unable to execute step: Data Joining 1
dfexec :: error :: aborted due to errors

Data source 1 contains 583,166 rows
Data source 2 contains 10,125,806 rows
```

You may be able to free up resources by increasing the efficiency of the job. Check the conditions of the Join and see if you are running a many-to-many Join. Also, check to see if there is a high number of NULL values in the two tables you are joining, which can also cause a huge join. You also need to check the memory limitations on the UNIX system where the Data Management Server process is running to ensure that there is enough room for Data Management Server to work properly.

By default, Data Management Server uses the `/tmp` directory for temporary files. You need to redirect the `TMPDIR` environment variable by issuing the `export TMPDIR=/new_temp_path` command. You can check this environment variable and change the settings with the following steps:

1. Stop the Data Management Server service.
2. From the UNIX shell, type `env | grep TMPDIR`. If results come back blank that means you do not have your `TMPDIR` redirected as an environment variable.
3. Type `export TMPDIR=/new_temp_path`.
4. Start the Data Management Server service.
5. Edit the `.profile` file and add `export TMPDIR=/new_temp_path` to the file for the user who starts Data Management Server.

For more information on stopping and starting the service, see [Data Management Server Service](#).

Errors Using the Java Plugin for UNIX

The `architect.cfg` file contains a setting called `java vm` that references the location of the Java™ Virtual Machine (JVM) DLL. If this setting is not configured properly, you will receive an error that the JVM could not be loaded.

Make sure you compile your JAVA code using a Java Development Kit of the same version or earlier than the JRE version you are running on your Data Management Server machine.

If the java classpath setting is incomplete, Architect or Data Management Server will report an error because the code could not be loaded. Check to make sure your code and any dependencies are accessible and specified in the classpath setting.

If the java classpath setting is empty, the only Java code that will be accessible to the Java Plugin are the examples that ship with Architect and Data Management Server.

Appendix A: Code Examples

The following content instructs you on how to create and connect to the DataFlux® Data Management Server (Data Management Server). Zip files are available with files for the examples. The `integrationserversamples.zip` file is located in the `Data Management Server\version` directory for Microsoft® Windows® operating system installations.

The DataFlux Web Service Definition Language (WSDL) file contains the set of definitions to describe the Web service. You can point directly to this file using either the directory path, such as `drive:\Program Files\DataFlux\Data Management Server\version\share\arch.wsdl`, or the URL, using the following syntax:

```
http://server_name:port/?wsdl
```

Using an XML command, you can edit and view the `arch.wsdl` file that is installed on your Data Management Server. Update the SOAP:address location to reflect the hostname and port number of the Data Management Server. For example:

```
<SOAP:address location="http://localhost:21036"/>
```

Additionally, you can view the WSDL file using a web browser. From this view, the value of SOAP:address location will reflect your actual hostname and port number.

```
<SOAP:address location="http://host_name:21036"/>
```

There are coding examples of these operations in each language listed below: Get Object List, Post Object, Delete Object, Get Architect Service Params, Execute Architect Service, Run Architect Job, Run Profile Job, Get Job Status, Get Job Log, Terminate Job, and Clear Log.

Java

Use `wscompile`, supplied with the Java™ Web Services Developer Pack, to build Java classes that wrap the Data Management Server interface. This creates all of the classes required to interface with Data Management Server for any application that has the ability to use these classes.

Examples

Following are examples using the Java classes constructed from the WSDL.

```
////////////////////////////////////  
// 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.dmc");
obj.setObjectType(ObjectType.fromString("ARCHSERVICE"));
String res = stub.deleteObject(obj);

////////////////////////////////////
// 4) Get Architect Service Params
////////////////////////////////////
GetArchitectServiceParamResponse resp;
FieldDefinition[] defs;
resp=stub.getArchitectServiceParams("MYJOB.dmc","");
// 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 Architect 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.dmc", 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 Architect 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.dmc", vals, "");

////////////////////////////////////
// 7) Run Profile Job
////////////////////////////////////
String res=stub.runProfileJob(
    "MYJOB.pfi",      /* Job Name */
    "",              /* Output file to create (not used in this case) */
    "repos",         /* Repository name to write results to */
    "New Report",    /* Report name to create */
    "Description",   /* Description of run */
    0,              /* Append to existing (false) */
    vals,           /* var/values */
    ""              /* reserved */
);

////////////////////////////////////
// 8) 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()

////////////////////////////////////
// 9) 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();

////////////////////////////////////
// 10) Terminate Job
////////////////////////////////////
String res=stub.terminateJob(jobId);

////////////////////////////////////
// 11) Clear Log
////////////////////////////////////
String res=stub.deleteJobLog(jobId);

```

C++

The client API consists of three header files and one .lib file. The headers include all necessary type enumerations. All required .dlls are provided within the DataFlux Data Management Studio installation. A connection handle should be initialized before use and freed by the terminate function when no longer needed.

```

////////////////////////////////////
// Imports
////////////////////////////////////
#include "arscli.h"
#include "acjob.h"
#include "acrta.h"

Also requires arscli11.lib

////////////////////////////////////
// INITIALIZATION
////////////////////////////////////
acj_handle_t *pHandle = acj_initialize(sServer, nPort);

////////////////////////////////////
// DESTRUCTION OF HANDLE at end of use
////////////////////////////////////
acj_terminate(pHandle);

```

```

////////////////////////////////////
// ERROR MESSAGES
////////////////////////////////////
const char *err_code, *err_text, *err_detail;
err_code = acj_get_error(pHandle, &err_text, &err_detail);

////////////////////////////////////
// 1) Get Object List example
////////////////////////////////////
int nNumJobs;
char **job_list;
job_list = acj_joblist(pHandle, RTARCHITECT /*ARCHITECT or PROFILE*/, &nNumJobs);

////////////////////////////////////
// 2) Post Object example
////////////////////////////////////
rc = acj_post_job(pHandle, "JOB_NAME", "FILE", RTARCHITECT/*ARCHITECT, PROFILE*/);

////////////////////////////////////
// 3) Delete Object
////////////////////////////////////
rc = acj_delete_job(pHandle, "JOB_NAME", RTARCHITECT/*ARCHITECT, PROFILE*/);

////////////////////////////////////
// 4) Get Architect Service Params
////////////////////////////////////
int nNumInputs, nNumOutputs;
const char *err_code, *err_text, *err_detail;
rc = acj_rt_io_info(pHandle, mJobName, &nNumInputs, &nNumOutputs);
int i, nColSize;
rta_data_type nColType;
const char *sColName;
rc = acj_rt_input_fieldinfo(pHandle, i, &sColName, &nColType, &nColSize);
rc = acj_rt_output_fieldinfo(pHandle, i, &sColName, &nColType, &nColSize);

////////////////////////////////////
// 5) Execute Architect Service
////////////////////////////////////
// Set up the input columns
int i, rc = 0;
CString sColName;

// This data is set when getting the parameter info
int *mColSizes new int[nNumInputs];
rta_data_type *mColTypes = new rta_data_type[nNumInputs];

//Loop and Load inputs FIELD info
rc = rta_set_infield_info(pHandle, i, sColName, mColTypes[i], mColSizes[i]);
//Loop and Add the input data
rc = rta_add_row(pHandle);
//For Each row add all the data for the columns/fields
rc = rta_set_data_value(pHandle, j, "VALUE");
// Run the test
rc = rta_run(pHandle);
// Get the number of output columns
int nNumCols;
nNumCols = rta_output_numfields(pHandle);
// Get the output column information
int nOutSize;
rta_data_type nOutType;

```

```

const char *sOutName;
for (i = 0; i < nNumCols; i++)
rc = rta_output_fieldinfo(pHandle, i, &sOutName, &nOutType, &nOutSize);
// Get the number of output rows
int nNumRows;
nNumRows = rta_output_numrows(pHandle);

// Get The output
const char *sOutVal;
for (i = 0; i < nNumRows; i++)
for (j = 0; j < nNumCols; j++)
sOutVal = rta_output_data(pHandle, j, i);
acj_terminate(pHandle);

////////////////////////////////////
// 6) Run Architect Job
////////////////////////////////////
int rc;
int mVarCount = 1;
acj_arch_var_value *mVarArray;
mVarArray = new acj_arch_var_value[mVarCount];

//LOAD ARRAY
CString sTemp = "Test Data";
mVarArray[0].var_name = new char[sTemp.GetLength() + 1];
strcpy(mVarArray[0].var_name, sTemp);
sTemp = "Test Value";
mVarArray[0].var_value = new char[sTemp.GetLength() + 1];
strcpy(mVarArray[0].var_value, sTemp);

char sJobID[ACJ_JOBID_SIZE];
CString sJobName = "JOB_NAME";
acj_job_type nType = ARCHITECT;

rc = acj_run_arch_job(pHandle, sJobName, mVarArray, mVarCount, sJobID);

////////////////////////////////////
// 7) Run Profile Job
////////////////////////////////////
int rc;
int mVarCount = 1;
acj_arch_var_value *mVarArray;
mVarArray = new acj_arch_var_value[mVarCount];

//LOAD ARRAY
CString sTemp = "Test Data";
mVarArray[0].var_name = new char[sTemp.GetLength() + 1];
strcpy(mVarArray[0].var_name, sTemp);
sTemp = "Test Value";
mVarArray[0].var_value = new char[sTemp.GetLength() + 1];
strcpy(mVarArray[0].var_value, sTemp);

char sJobID[ACJ_JOBID_SIZE];
CString sJobName = "JOB_NAME";
// REPORT FILE
rc = acj_run_prof_job(pHandle, sJobName, "FileName", 0,
1/*Append - 1, Truncate - 0*/, 0,
"Description", mVarArray, 1, sJobID);

// Repository
rc = acj_run_prof_job(pHandle, sJobName, 0, "ReposName",

```



```

1/*Append - 1, Truncate - 0*/, "ReportName",
"Description", mVarArray, 1, sJobID);

////////////////////////////////////
// 8) Get Job Status
////////////////////////////////////
int nNumStats;
int rc = acj_get_job_status(pHandle, ""/*or "JobID"*/, &nNumStats);

acj_job_type nType;
char *sName, *sJobID, *sStatus;

for (int i = 0; i < nNumStats; i++)
rc = acj_get_job_status_item(pHandle, i, &sName, &sJobID, &nType, &sStatus);

////////////////////////////////////
// 9) Get Job Log
////////////////////////////////////
char sLogFile[MAX_PATH];
GetTempFileName(dfReadIniFile("Environment", "WorkingPath"), "ISM", 0, sLogFile)

int rc = acj_get_job_log(pHandle, "JOBID", sLogFile);

////////////////////////////////////
// 10) Terminate Job
////////////////////////////////////
int rc = acj_terminate_job(pHandle, "JOBID");

////////////////////////////////////
// 11) Clear Log
////////////////////////////////////
int rc = acj_delete_job_log(pHandle, "JOBID");

```

C#

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

```

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

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

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

```

```

////////////////////////////////////
// 2) Post Object example
////////////////////////////////////
DQISServer.ObjectDefinition def = new DQISServer.ObjectDefinition();
def.objectName = "MYJOB";
def.objectType = DQISServer.ObjectType.ARCHSERVICE;

// Grab Bytes from a job file
byte[] data = new byte[short.MaxValue];
FileStream fs = File.Open(@"c:\Develop\SoapUser\DISTESTRT.DMC",
    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 = "MYJOB";
def.objectType = DQISServer.ObjectType.ARCHSERVICE;

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

////////////////////////////////////
// 4) Get Architect 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 Architect 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 Architect 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) Run Profile Job
////////////////////////////////////
DQISServer.SendRunProfileJobRequestType req = new
DQISServer.SendRunProfileJobRequestType();
DQISServer.GetRunProfileJobResponseType resp;

req.jobName = "JOB_NAME";
req.reportName = "REPORT_NAME";
// use this: req.repositoryName = "REPOSNAME";
// or this:
req.fileName = "FILE_NAME";

req.description = "DESCRIPTION";

req.append = 0;//No - 0; Yes - 1

resp = mService.RunProfileJob(req);

string jobId = resp.jobId;

////////////////////////////////////
// 8) 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;

////////////////////////////////////
// 9) 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;

////////////////////////////////////
// 10) Terminate Job
////////////////////////////////////
DQISServer.SendTerminateJobRequestType req = new
DQISServer.SendTerminateJobRequestType();
DQISServer.GetTerminateJobResponseType resp;
req.jobId = "SOMEJOBID";

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

////////////////////////////////////
// 11) Clear Log
////////////////////////////////////
DQISServer.SendDeleteJobLogRequestType req = new
DQISServer.SendDeleteJobLogRequestType();
DQISServer.GetDeleteJobLogResponseType resp;
req.jobId = "SOMEJOBID";

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

```

Appendix B: Security Policy Planning and Examples

Security Policy Planning

A well-planned security model allows the Data Management Server security administrator (admin) to control access to the application. Data Management Server offers several security tools, allowing the administrator to work with your existing security policy. As a resource on your network, Data Management Server usage can be defined based on your security model, which in turn is based on usage policy, risk assessment, and response. Determining user and group usage policies prior to implementation helps you minimize risk and expedite deployment.

Risk Assessment — Security policies are inevitably a compromise between risk and necessary access. Users must access the application and data in order to perform necessary tasks, but there is associated risk when working with information, particularly confidential data. Consider the risks of compromised (unauthorized views or lost) data. The greater the business, legal, financial, or personal safety ramifications of compromised data, the greater the risk.

Usage Policy — Determine usage policy based on risk assessment. Take into account individual and group roles within your organization. What policies are already in place? Do these users or groups already have access to the data used by Data Management Server? Are they Data Management Studio users? Generally, users will fall into one of the following categories: administrators, power or privileged users, general users, partners, and guests or external users. The approach to *deny all, allow as needed* will help you to implement security from the top down. New users should have restricted access. Access for administrators and power users could then be conferred manually or through explicit group authorizations.

Security Response — Consider establishing a security response policy. If you have a security response team, specify how they are to respond to and report violations of security policy. Consider training all users on acceptable use prior to deployment of Data Management Server.

Data Management Server Security Examples

There are two types of security available with DataFlux® Data Management Server (Data Management Server), IP-based security, and Data Management Server Security Manager. IP-based security, configured in the `dfexec.cfg` file, controls user access by IP address. For more information, see [IP-Based Security](#). The Data Management Server Security Manager application is part of the Data Management Server. Through the Security Manager interface, user access can be controlled based on user, group, and job level authorizations. These security tools can be used separately or together. The following scenarios employ different types of security:

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

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

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

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

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

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

Client request:

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

Server response:

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

Client request:

```
GET /private/index.html HTTP/1.0
Host: localhost
Authorization: Basic QWxhZGRpbjpvYVUHNlc2FtZQ==
```

Server response:

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

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

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

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

Appendix C: Using Configuration Settings

Settings

The following sections describe how to use configuration settings in your Data Management Server environment.

Using Configuration Settings to Pre-Load Services

Data Management Server can preload selected services on startup. This is helpful if you typically use the same services each time you run Data Management Server and would like to have these services available as soon as Data Management Server is running.

There are two configuration directives available that cause Data Management Server to preload services; these can be set by the Data Management Server administrator:

- `dfsvc preload all = count`
- `dfsvc preload = count:name_of_service count:name_of_service ...`

where *count* specifies the number of pre-load counts and *name_of_service* indicates the name of the service element. This may include the directory where the service is located.

The two formats can work independently or together, depending on how you configure them.

Pre-loading All Services

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

For example, `dfsvc preload all = 2` causes Data Management Server to preload two instances of each service that is available, including those found in subdirectories.

Pre-loading One or More Specific Services

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

For example, `dfsvc preload = 2:abc.dmc 1:subdir1\xyz.dmc` loads two counts of `abc` service, and one count of `xyz` service, which is located in the `subdir1` subdirectory.

Complex Configurations

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

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

```
dfsvc preload all = 2
```

```
dfsvc preload = 2:svc1.dmc -1:subdir1\svc2.dmc -2:svc3.dmc
```

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

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

Some important points to remember:

- Data Management Server attempts to preload a single instance of all requested services before trying to preload more instances (if more than one instance is specified).
- The service element can include the path to the service, relative to the root of the services directory. For example, `1:subdir1\svc2.dmc` specifies one instance of service `svc2.dmc`, which is located in the `subdir1` subdirectory.
- Count can be a negative value. This is meaningful only when both configuration directives are used together.
- Pre-loading stops when Data Management Server has attempted to preload all required instances (successfully or not), or if the limit on the number of services has been reached. The limit can be specified by `dfsvc max num =`, and will default to 10 if not specified.

Using Configuration Settings for Multi-Threaded Operation

Two configuration directives control whether the servers run:

- **svr run dmserver = [yes/no]** — The default is yes.
- **svr run wlp = [yes/no]** — The default is no.

There are three additional configuration directives that determine how the thread pool operates:

- **svr max threads = *#_of_threads*** — If WLP server is to run, at least two threads are used; if SOAP server is to run, at least four threads are used; Data Management Server automatically adjusts this value to the required minimum if the configured value is too low.
- **svr max idle threads = *#_of_threads*** — Will always be at least 1. This directive should be treated as an advanced configuration, and should be used only when needed to troubleshoot performance problems.
- **svr idle thread timeout = *#_of_microseconds*** — Defaults to 5 seconds if not set or if set to less than 1 microsecond. This directive should be treated as an advanced configuration, and should be used only when needed to troubleshoot performance problems.

Glossary

A

ACE

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

ACL

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

API

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

D

DAC

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

DPV

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

DSN

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

L

LACS

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

M

MMC

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

O

ODBC

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

Q

QAS

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

QKB

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

R

RDI

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

S

SERP

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

SOA

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

SOAP

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

U

USPS

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