

# DataFlux Federation Server Administrator's Guide



YOUR DATA.  
YOUR BUSINESS.  
ONE SOLUTION.



This page is intentionally blank



# DataFlux Federation Server

## Administrator's Guide

---

Version 2.1.3

November 21, 2011

This page is intentionally blank

# Contact DataFlux

## DataFlux Corporate Headquarters

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

## DataFlux United Kingdom

Tel: +44 (0) 20 3176 0025  
Fax: +44 (0) 20 3411 8382  
Enterprise House  
1-2 Hatfields  
London  
SE1 9PG  
United Kingdom

## DataFlux Germany

Tel: +49 (0) 69 66 55 42 04  
In der Neckarhelle 162  
69118 Heidelberg  
Germany

## Technical Support

Phone: 1-919-531-9000  
Email: [techsupport@dataflux.com](mailto:techsupport@dataflux.com)  
Web: <http://dataflux.com/MyDataFlux-Portal.aspx>

## Documentation Support

Email: [docs@dataflux.com](mailto:docs@dataflux.com)

## DataFlux West

Tel: (818) 906-7638  
Fax: (818) 907-6012

15300 Ventura Boulevard, Suite 523  
Sherman Oaks, CA 91403  
USA

## DataFlux France

Tel: +33 (0) 4 72 91 31 42

Immeuble Danica B  
21, avenue Georges Pompidou  
69003 Lyon  
France

## DataFlux Australia

Tel: +61 2 9428 0553  
300 Burns Bay Road  
Lane Cove, NSW 2066  
Australia

# Legal Information

Copyright © 1997 - 2011 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.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

[DataFlux Legal Statements](#)

[DataFlux Solutions and Accelerators Legal Statements](#)

## DataFlux Legal Statements

### Apache Portable Runtime License Disclosure

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

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

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

### Apache/Xerces Copyright Disclosure

The Apache Software License, Version 3.1

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

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

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

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

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

4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [apache@apache.org](mailto: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.

## **VMware**

DataFlux Corporation LLC technical support service levels should not vary for products running in a VMware® virtual environment provided those products faithfully replicate the native hardware and provided the native hardware is one supported in the applicable DataFlux product documentation. All DataFlux technical support is provided under the terms of a written license agreement signed by the DataFlux customer.

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

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

## **Solutions and Accelerators Legal Statements**

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

## **Apache**

This product may contain software technology licensed from Apache.

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

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

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

## **Creative Commons Attribution**

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

## Degrafa

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

Copyright © 2008-2010 Degrafa. All rights reserved.

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

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

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

## Google Web Toolkit

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

## JDOM Project

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

## OpenSymphony

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

## Sun Microsystems

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

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

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

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

## Java Toolkit

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

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

# Table of Contents

<b>Introduction .....</b>	<b>1</b>
Accessibility .....	1
Conventions Used In This Document .....	1
Reference Documentation.....	2
Supplemental Documentation .....	2
<b>Overview .....</b>	<b>3</b>
Services Provided by the DataFlux Federation Server .....	4
Components Provided by the Federation Server .....	7
<b>System Requirements .....</b>	<b>9</b>
Supported Operating Systems and Required Host Hardware .....	9
Enable Streams on HP-UX .....	10
Supported Databases .....	11
Supported Database Clients .....	11
<b>Installing DataFlux Federation Server .....</b>	<b>12</b>
Windows .....	12
UNIX.....	12
<b>Installing and Configuring the Driver for SAP .....</b>	<b>14</b>
Driver for SAP Requirements.....	14
Authorization Profiles .....	15
Setting up the Driver for SAP .....	16
Installing SAP Components .....	17
<b>Configuring the UNIX Environment .....</b>	<b>24</b>
Set the LANG Environment Variable .....	24
Configuring the Federation Server Drivers .....	24
<b>Configuring DataFlux Federation Server.....</b>	<b>31</b>
Configuring the Federation Server Service .....	31
Configuring Connections.....	32
Configuring Licensing .....	34
Managing Configuration Files and Scripts.....	35

<b>DataFlux Federation Server Concepts.....</b>	<b>43</b>
Data Services.....	43
Data Source Names (DSNs) .....	45
Users, Groups, and Roles .....	49
Authorization and Security.....	50
Database Functionality Supported by Drivers .....	54
<b>SAS Data Set Reference .....</b>	<b>58</b>
Catalogs and Schemas .....	58
Data Types .....	59
Understanding the Driver for BASE .....	63
Database Functionality Support for SAS Data Sets.....	64
Data Service Connection Arguments for SAS Data Sets .....	65
Data Types for SAS Data Sets .....	67
SAS Column Labels for SAS Data Sets.....	69
Optimize Performance for SAS Data Sets.....	69
Indexes for SAS Data Sets.....	69
Audit Trail for SAS Data Sets .....	70
Integrity Constraints for SAS Data Sets.....	70
Generation Data Sets for SAS Data Sets.....	70
Cross-Environment Data Access for SAS Data Sets .....	71
Identifier Case Sensitivity for SAS Data Sets.....	71
Security for SAS Data Sets .....	73
Numeric Column Length for SAS Data Sets .....	73
Naming Conventions and Attributes for SAS Data Sets .....	74
National Language Support.....	75
TSSQL Statement Table Options .....	75
<b>SAP Reference.....</b>	<b>87</b>
Understanding the Driver for SAP .....	87
Data Service Connection Arguments for SAP .....	87
Data Types for SAP .....	91
<b>DB2 Reference .....</b>	<b>93</b>

Understanding the Driver for DB2 .....	93
Database Functionality Support for DB2 .....	93
Data Service Connection Arguments for DB2.....	94
Data Types for DB2 .....	97
Load Data by Using Bulk Operations for DB2.....	99
Naming Conventions for DB2 .....	100
TSSQL Statement Table Options.....	100
<b>MySQL Reference .....</b>	<b>110</b>
Understanding the Driver for MySQL .....	110
Database Functionality Support for MySQL .....	110
Data Service Connection Arguments for MySQL.....	111
Data Types for MySQL.....	115
Naming Conventions for MySQL .....	117
TSSQL Statement Table Options.....	118
<b>ODBC Reference .....</b>	<b>120</b>
Understanding the Driver for ODBC .....	120
Database Functionality Support for ODBC.....	120
Data Service Connection Arguments for ODBC .....	121
Data Types for ODBC .....	124
Naming Conventions for ODBC.....	126
<b>Oracle Reference.....</b>	<b>127</b>
Understanding the Driver for Oracle.....	127
Database Functionality Support for Oracle .....	127
Data Service Connection Arguments for Oracle.....	128
Data Types for Oracle .....	131
Load Data by Using Bulk Operations for Oracle.....	134
Naming Conventions for Oracle .....	134
TSSQL Statement Table Options.....	135
<b>Teradata Reference .....</b>	<b>140</b>
Understanding the Driver for Teradata .....	140
Database Functionality Support for Teradata.....	140

Data Service Connection Arguments for Teradata .....	141
Data Types for Teradata .....	143
Naming Conventions for Teradata .....	145
TSSQL Statement Table Options .....	145
<b>Technical Support .....</b>	<b>153</b>
Best Practices .....	153
Troubleshooting the DataFlux Federation Server Deployment .....	155
<b>Appendix A: TSSQL DDL Statements.....</b>	<b>158</b>
GENERIC OPTIONS Syntax .....	159
ALTER GENERIC OPTIONS Syntax.....	159
ALTER SERVER Statement .....	160
CREATE DATA SERVICE Statement .....	161
DROP DATA SERVICE Statement .....	162
ALTER DATA SERVICE Statement .....	163
CREATE CATALOG Statement .....	165
DROP CATALOG Statement .....	166
ALTER CATALOG Statement .....	166
CREATE SCHEMA Statement .....	167
DROP SCHEMA Statement .....	168
ALTER SCHEMA Statement .....	169
CREATE DSN Statement .....	170
DROP DSN Statement .....	172
ALTER DSN Statement .....	172
DROP AUTHID Statement .....	173
GRANT and DENY Statements .....	174
REVOKE Statement.....	176
<b>Appendix B: Federation Information Views .....</b>	<b>178</b>
Visibility of Information Views .....	178
AUTHORIZATION_IDENTIFIERS View .....	179
IDENTITY View .....	180
CATALOG_PRIVILEGES View .....	180

CATALOGS View .....	181
DATA_SERVICES View.....	181
SCHEMAS View .....	182
DS_PRIVILEGES View .....	182
SCHEMA_PRIVILEGES View.....	183
DSN_PRIVILEGES and EFFECTIVE_DSN_PRIVILEGES Views .....	184
DATA_SOURCE_NAMES View .....	186
DSN_CONTENT View .....	187
CONFIG_DSNS View .....	187
CONFIG_SCHEMAS View .....	187
CONFIG_DATA_SERVICES View.....	188
CONFIG_CATALOGS View .....	188
PRIVILEGES and EFFECTIVE_PRIVILEGES Views.....	188
<b>Appendix C: Code Example .....</b>	<b>191</b>
SAS_CACHED_CATALOG File.....	191
<b>Glossary .....</b>	<b>193</b>

# Introduction

This section provides basic information about the DataFlux® Federation Server (Federation Server) product and documentation.

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

## Accessibility

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

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

## Conventions Used In This Document

This document uses several conventions for special terms and actions.

### Typographical Conventions

The following typographical conventions are used in this document:

Typeface	Description
<b>Bold</b>	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, <b>Start &gt; Programs &gt; DataFlux &gt; Data Management Studio <i>version</i> &gt; License Manager</b> .
::=	This symbol can be interpreted as "consists of" or "is defined as".



## Path Conventions

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

### Windows XP

*drive:\Program Files\DataFlux\FederationServer\version*

### Windows 7

32-bit – *drive:\Program Files (x86)\DataFlux\FederationServer\version*

64-bit – *drive:\Program Files\DataFlux\FederationServer\version*

### UNIX®

*/opt/dataflux/dfs*

## Reference Documentation

This document may reference other DataFlux documentation, including:

*DataFlux Authentication Server Administrator's Guide*

*DataFlux Authentication Server User's Guide*

*DataFlux Federation Server User's Guide*

*DataFlux Federation Server TSSQL Reference Guide*

*DataFlux Data Management Server Administrator's Guide*

*DataFlux Data Management Server User's Guide*

*DataFlux Data Management Studio Online Help*

## Supplemental Documentation

The following SAS® documentation supplements this Federation Server documentation:

*SAS 9.2 Language Reference: Concepts*

*SAS 9.2 National Language Support (NLS): Reference Guide*

# Overview

The DataFlux Federation Server (Federation Server) is a data server that provides scalable, threaded, multi-user, and standards-based data access technology in order to process and seamlessly integrate data from multiple data sources. The server acts as a hub that provides clients with data by accessing, managing, and sharing SAS® data as well as several third-party relational databases.

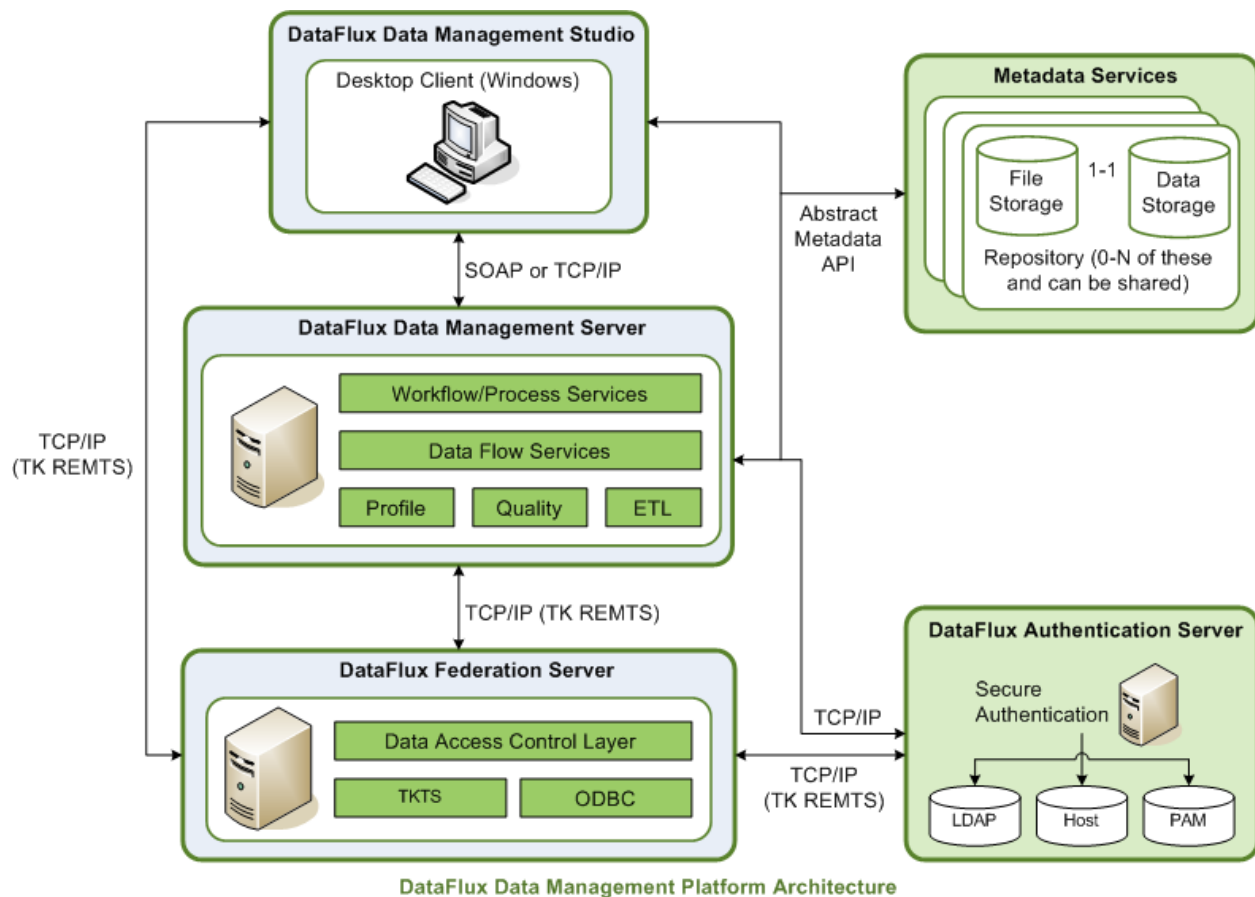
The Federation Server provides the following key benefits:

- Both user scalability and processing scalability to improve performance.
- Threaded data access technology that enhances enterprise intelligence and analytical processes.
- Multi-user services that enable multiple clients to access the same data concurrently.
- Access to multiple types of data, which removes the need for a client to connect to one server for some data and to another server for other data. The Federation Server supports several third-party relational databases including DB2®, MySQL®, Oracle®, ODBC databases, SAP®, and Teradata®.
- A standards-based interface for Structured Query Language (SQL), which defines the data access language for the server. An application creates, requests, and manipulates data with SQL statements. SQL can be submitted by using Federation Server TSSQL (TSSQL), which supports the ANSI SQL:1999 syntax, or by using the SQL dialect that is implemented by a third-party database.
- Federated queries that reference multiple tables from different data sources, thereby joining disparate data.
- Security services that ensure that both the server and its data are protected against unauthorized access. The Federation Server supports authorization processes and other security features such as encryption.

A client connects to the Federation Server and submits requests in the form of SQL statements. SQL is the data access language for the server. The server manages the requests and delivers the results to the client. A client can submit a simple request that accesses a specific table. A client can also submit a federated query that references multiple tables from different data sources, thereby joining disparate data.

By supporting industry standards and functionality, the Federation Server is designed to support data integration and business intelligence solutions. The server provides a single SQL language interface that is data source-independent to clients of distributed enterprise data.

The Federation Server is a product which can be used by DataFlux Data Management Studio (Studio) and DataFlux Data Management Server. The following figure illustrates the DataFlux Data Management Platform architecture, with the Federation Server as a single point of access to provide multiple client applications with data from different sources:



For information on Studio and how to configure Federation Server connections, see the *DataFlux Data Management Studio Online Help*. For information on DataFlux Data Management Server and how to configure Federation Server connections, see the *DataFlux Data Management Server User's Guide*.

## Services Provided by the DataFlux Federation Server

This section contains information about the services provided by the Federation Server.

### Data Access Technology

The data access technology that is provided by the Federation Server consists of a set of run-time components that provide a scalable, threaded, multi-user, and standards-based way to process and seamlessly integrate data from multiple data sources. The components provide the data access services that are required by business intelligence and analytical processes.

### Threaded Services

Threads are an integral part of a high-performance, scalable system, and they are one of the main features of the Federation Server data access technology. Most threaded functionality can be further boosted in an environment in which multiple processors work in

parallel. However, performance boosts can also be obtained with multi-threaded processes on a single processor machine.

A threaded service is a method of processing that divides a large job into several smaller jobs that can be executed in parallel. Threaded services control and execute requests by using multiple threads to increase data throughput. A thread is a single path of execution of a process in a single CPU. A thread can also be thought of as a basic unit of program execution in a thread-enabled operating environment. In a symmetric multiprocessing (SMP) environment, which uses multiple CPUs, multiple threads can be spawned and processed simultaneously. Regardless of whether there is one CPU or many, each thread is an independent flow of control that is scheduled by the operating system.

The Federation Server provides threaded services that execute multiple user requests in parallel. Here are examples:

- Each connection to the Federation Server is managed on a separate thread, which enables multiple users to execute requests in parallel and reduces the probability of a user request being blocked while other user requests are processed.
- Complex requests (or large individual requests) are separated into units of work that are then executed in parallel. For example, filtering operations that require scanning large tables can be processed in parallel, and operations such as sorting can be processed by dividing the result set into subsets, sorting each subset in parallel, and then merging the sorted subsets into the final result set.
- Threading is also used to return result sets on multiple threads. For example, a client can submit a TSSQL query and request that the result set be returned in mutually exclusive partitions. Each partition is returned on a separate thread. The management of the threads is shared between the Federation Server driver (which creates the threads) and the client (which cleans up the threads).

In addition to threaded services, some data services provide threaded I/O, which further enhance performance. For more information, see [Threaded I/O Processing](#).

## Multi-User Services

Multi-user services enable multiple clients to access the same data concurrently. If the data source supports this capability, the Federation Server enables two or more clients to write to the same table at the same time without destroying or losing updates, which is referred to as concurrent update access.

The Federation Server uses Integrated Object Model (IOM) technology. IOM technology is a set of object-based interfaces to features or services. The technology enables application developers to use industry-standard programming languages, programming tools, and communication protocols to develop client programs that access these services on IOM servers.

A multi-user environment automatically ensures data protection during concurrent updates. The data services support concurrent updates by locking the data that is being updated and releasing the lock when updates are complete. This prevents loss of data or loss of updates that are due to simultaneous updates.

For more information, see [Locking](#).

## Performance

The Federation Server integrates both user scalability and processing scalability to provide increased performance. That is, the Federation Server supports the following performance capabilities:

- In a multi-user environment, the server automatically scales to the number of concurrent users.
- The server provides rapid access to large amounts of data.
- The server is available under many 64-bit operating environments, which enables the server to scale in-memory processes.
- The server provides application-based, high-performance data reading by supporting a variety of cursor types, multi-row fetch capabilities, and positioned update of result sets.

## Data Storage Support

The Federation Server provides access to several types of data, which removes the need for an application to connect to one server for some data and another server for other data. The Federation Server supports SAS® data sets, SAP®, and third-party relational databases as data sources. Relational databases include:

- IBM® DB2
- MySQL
- Oracle
- Microsoft® SQL Server®
- Teradata

By supporting several data sources, the Federation Server gives you the flexibility to configure data storage based on specific needs. You choose the type of data storage that is most appropriate for the particular needs of an application, based on functionality that is provided by each data sources. Here are examples:

- If the application needs row-level locking and transaction support, use DB2 or Oracle.
- For data types such as BIGINT and VARCHAR, use data sources such as DB2 or Oracle.
- For the most effective and efficient storage for use with SAS analytical and business intelligence software, use SAS data sets.

For more information about the supported data sources and the functionality they provide, see [Data Services](#).

## Standards-Based Interface for SQL

The Federation Server provides a standards-based interface for the SQL, which defines the data access model for the server. That is, an application creates, requests, and manipulates data by submitting SQL statements.

An application can submit SQL statements by using TSSQL, which is the dialect supported by the SQL processor built into the Federation Server. For more information, see [TSSQL](#).

## Security

The Federation Server security services ensure that both the server and its data are protected against unauthorized access. The Federation Server supports configurable authorization processes and other security features, including encryption. In addition, the Federation Server provides the ability to control access to SAS data sets that are placed under exclusive control of the server.

For more information, see [Authorization and Security](#).

## Components Provided by the Federation Server

The Federation Server consists of a set of components that provide the functionality that is required by data integration, business intelligence, and analytic processing. For example, the Federation Server provides several types of drivers and two interfaces that you use to connect to the server. The following topics describe each component.

### Federation Server Drivers

A Federation Server driver is software that interacts with a data source in order to read and write to proprietary file formats. Each data source that is supported by the Federation Server has a driver that talks to the data service in its own language in order to resolve data access requests and to manage the physical file or database table. For example, to process an SAP table, an application uses the DataFlux Driver for SAP, and to process an Oracle table, an application uses the DataFlux Driver for Oracle.

The Federation Server driver connects to and disconnects from the data service, submits SQL statements to the data service for execution, and sends data to and retrieves data from the data service. That is, a Federation Server driver receives an SQL expression as input and returns a result as output. Each Federation Server driver supports the database functionality of the underlying data source.

Federation Server drivers enable the Federation Server to access files that are in a particular format.

For more information about data services and the Federation Server drivers, see [Data Services](#).

### Language Drivers

The language drivers implement the Federation Server languages by processing a request and sending the parsed query to the appropriate Federation Server driver that satisfies the request and returns the result. The multi-threaded languages provide a powerful way to create and query data.

### TSSQL

TSSQL is the implementation of SQL that the Federation Server uses to access relational data. TSSQL is designed to be ANSI SQL:1999 core compliant with some extensions.

For applications, TSSQL provides a common SQL syntax across all data services. That is, TSSQL is a vendor-neutral SQL dialect that accesses data from various data services without having to submit queries in the SQL dialect that is native to the data service. In addition, a single TSSQL query can target data in several data services and can return a single result set. When possible, TSSQL queries are optimized with multi-threaded algorithms to resolve large-scale operations.

The TSSQL language driver parses TSSQL requests, and then sends the parsed query to the appropriate Federation Server driver to determine whether the functionality can be handled by the data service.

For the DataFlux Driver for SAS BASE Library, the parsed TSSQL request is interpreted “as is” because TSSQL is the native SQL dialect for the data service.

For the relational database Federation Server drivers, if the data service supports the functionality, then the TSSQL request is translated to the data service’s native SQL dialect. By enabling as much functionality as possible to be handled by the data service, performance is enhanced. However, if a data service does not support the requested functionality, then the TSSQL language driver attempts to compensate by completing the balance of the operation.

For complete TSSQL statement reference, see the *DataFlux Federation Server: TSSQL Language Reference*.

# System Requirements

## Supported Operating Systems and Required Host Hardware

Operating System	Bits	OS Versions and Patches	Memory	Min. Disk Space*
AIX®	64	IBM® AIX 5.3 Technology Level 6 or AIX 6.1, with the runtime package <code>xlcpp.rte.10.1.0.aix.base</code> . Note that you can run the 32-bit system if required by third-party software.	96MB per concurrent user	694MB
HP-UX®	64	HP-UX 11i version 2 or 3, operating system release identifier <code>B.11.23</code> and above, with June 2007 patch bundle. If you authenticate with the Pluggable Authentication Module, install patches <code>PHCO_40361</code> and <code>PHCO_4036</code> , which make the Itanium patches 40360 and 40361 available on PA-RISC. Also install the Atomic APIs from the AtomicAPI optional software pack. See also <a href="#">Enable Streams on HP-UX</a> .	96MB per concurrent user	793MB
HP-UX Itanium® Processor Family Architecture	64	HP-UX 11i version 2 or 3, operating system release identifier <code>B.11.23</code> and above, with the June 2007 patch bundle and the <code>PHKL_36853</code> patch. If you authenticate with the Pluggable Authentication Module, install patches <code>PHCO_40360</code> and <code>PHCO_40361</code> . Also install the Atomic APIs from the AtomicAPI optional software pack. See also <a href="#">Enable Streams on HP-UX</a> .	96MB per concurrent user	1063MB
Red Hat Linux®	32 or 64	Red Hat Enterprise Linux 5.3, with version 2.6.11 or higher of the Linux kernel (see also the kernel information in the <a href="#">Red Hat Knowledge Base</a> ). Install all default packages. For x64, install both the 32-bit and 64-bit versions of the <code>libXp.so</code> library, which is included in the <code>libXp</code> package. Install the RPM package <code>compat-libstdc++-33-3.2.3-61</code> . The Unicode libraries depend upon the installation of a compatible standard C++ library, which is located in <code>/usr/lib/libstdc++.so.5</code> and/or <code>/usr/lib64/libstdc++.so.5</code> . The Native Posix Thread Library is supported. Linux threads are not supported. <code>glibc 2.4</code> is required during the build.	64MB plus 8MB per concurrent user	722MB (32) or 784MB (64)
SuSE Linux	32 or 64	Open SuSE 10.2 or 11.0 with version 2.6.11 or higher of the Linux kernel. SuSE 11.0 has better thread support. Install all default packages. For SuSE 10.2, install RPM package <code>compat-libstdc++-5.0.7-22.2</code> . The Unicode libraries depend upon the installation of a compatible standard C++ library, which is located in <code>/usr/lib/libstdc++.so.5</code> and/or <code>/usr/lib64/libstdc++.so.5</code> . The Native Posix Thread Library is supported. Linux threads are not supported.	64MB plus 8MB per concurrent user	722MB (32) or 784MB (64)



Operating System	Bits	OS Versions and Patches	Memory	Min. Disk Space*
		glibc 2.4 is required during the build.		
Solaris®	64	Solaris S64 or SAX, Version 9 or Version 10 Update 1 and higher, with patch 120037-09 for LDAP authentication.	96 MB per user	883MB
Windows®	32 or 64	<p>Microsoft® Windows Server 2008 Standard Edition</p> <p>Microsoft Windows Server 2008 Enterprise Edition</p> <p>Microsoft Windows Server 2008 Datacenter Edition</p> <p>Microsoft Windows Server 2003, Standard Edition updated with Service Pack 1*</p> <p>Microsoft Windows Server 2003, Enterprise Edition updated with Service Pack 1*</p> <p>Microsoft Windows Server 2003, Datacenter Edition updated with Service Pack 1*</p> <p>* Install this update:  <a href="http://www.microsoft.com/downloads/details.aspx?familyid=17c36612-632e-4c04-9382-987622ed1d64&amp;displaylang=en">http://www.microsoft.com/downloads/details.aspx?familyid=17c36612-632e-4c04-9382-987622ed1d64&amp;displaylang=en</a></p> <p>Microsoft Windows Server 2003 for x64 systems, Standard Edition</p> <p>Microsoft Windows Server 2003 for x64 systems, Enterprise Edition</p> <p>Microsoft Windows Server 2003 for x64 systems, Datacenter Edition</p> <p>Microsoft Windows XP Professional, updated with Service Pack 2</p> <p>Microsoft Windows Vista®, Enterprise, Business, and Ultimate Editions</p> <p>Microsoft Windows XP Professional for x64 systems</p> <p>Microsoft Windows Vista for x64 systems, Enterprise, Business, and Ultimate Editions</p>	1GB plus 1GB swap space. For XP, 512MB plus 512MB swap space.	830MB

\* An additional 30MB of disk space is required during installation for temporary storage.

## Enable Streams on HP-UX

Follow these steps to ensure that Streams is installed and enabled on HP-UX:

1. The HP-UX Streams product is generally installed by default. Issue the following command to verify that Streams has been installed:  

```
/usr/sbin/swlist -l product | grep Streams
```
2. If necessary, install Streams from the HP-UX installation media. You will need to log in as a Superuser.
3. To ensure that Streams is enabled, issue the following command  

```
/usr/sbin/kctune -v streampipes
```

4. If the current value of the `streampipes` variable is 1, then Streams is enabled and this procedure is complete.
5. If the current value is 0, issue the following command to change the `streampipes` variable:  



```
/usr/sbin/kctune streampipes=1
```
6. Restart HP-UX to enable Streams.

## 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*.

## Supported Database Clients

The following table lists the client versions for the native drivers:

Database Client	Native Driver
DB2® V8.2 Fixpack 9 client and higher DB2 V9.1 Fixpack 4 client and higher  <b>Note:</b> The Windows 7 platform requires the DB2 V9.5 client.	DB2
MySQL® Version 5 and higher	MySQL
Oracle® 10g client and higher  <b>Note:</b> The SAX platform supports the Oracle 11g client.	Oracle
64-bit SAP® Unicode, Release 7.10 SAP Kernel Release 4.6C or higher	SAP
TTU 12 and higher (client and utilities, including TPT)	Teradata®

# Installing DataFlux Federation Server

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

Download the DataFlux Federation Server from the download section of the DataFlux MyDataFlux Portal at <http://www.dataflux.com/MyDataFlux-Portal>.

## Windows

To install Federation Server using the installation wizard, complete the following steps:

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

```
dmpversion-federation-winnumber.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 **Select Components** window, select **ODBC Drivers**, if it is not already selected by default, and click **Next**.
5. In the **Set System User** window, accept the default user ID as the primary system user for the installation or provide a new user ID, and then click **Next**.
6. In the **Authentication Server Location** window, specify the hostname and port number for the Authentication Server, and then click **Next**.
7. In the **Licensing** window, accept the default licensing method and location or select a new method and location, and then click **Next**.
8. 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**.

9. 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\FederationServer\version
```

## UNIX

To install Federation Server using the command line, complete the following steps:

1. Copy the Federation Server installation and `README.txt` file that corresponds to your operating system to an accessible directory, for example, AIX®, HP-UX, Linux, or Solaris™.

2. At the command prompt, connect to the location where you are loading Federation Server.
3. Specify the directory where you will be loading Federation Server, and navigate to that directory.
4. Enter the following command to uncompress the installation file. Replace *path\_to* in the command with the directory where you copied the installation file:

```
gzip -c -d path_to/dmpversion-fed-operating_system.tar.gz | tar xvf -
```
5. Run the installation program by typing:

```
perl dfs/install.pl
```
6. Accept the default user ID who will be the primary administrator for the Federation Server or specify a different user ID and press **Enter**.
7. Accept the default hostname of the Authentication Server or specify a different hostname and press **Enter**.
8. Accept the default port number for the Authentication Server or specify a different port number and press **Enter**.
9. Accept the default licensing method or specify a different licensing method and press **Enter**.
10. Accept the default licensing location or specify a different licensing location and press **Enter**.
11. Once the installation completes, you can view the Release Notes by reading the `ReleaseNotes.txt` file.

The default installation directory for the server is:

```
/opt/dataflux/dfs
```

# Installing and Configuring the Driver for SAP

Installing the DataFlux® Driver for SAP (Driver for SAP) involves several steps that you must complete in the appropriate sequence. Review the system requirements and authorization profiles, set up the Driver for SAP, and then install the SAP® components on the SAP system and the DataFlux Federation Server (Federation Server).

After installing the Driver for SAP, you must complete additional steps to configure it to be used by the Federation Server.

## Driver for SAP Requirements

The following section identifies the system requirements for a Driver for SAP that will be used by the Federation Server. Once the Driver for SAP has been installed, verify that the following requirements have been met:

### SAP System

The SAP Kernel Release 4.6C or higher is required.

The Driver for SAP for Windows® and UNIX® requires the 64-bit SAP Unicode RFC library, Release 7.10, which is provided by SAP AG. Refer to SAP Note 413708 for the current version, download and installation instructions.

### SAPGUI

During the installation of the SAP components, a SAPGUI is required.

### User IDs

A SAP user ID and password is required. The user ID must have appropriate authorizations to access data and use communication methods. For more information about customizing the authorization, see [Authorization Profiles](#).

To install and run Driver for SAP, the following SAP user IDs are required:

- **RFC user** — This is a SAP user ID that is used for the communication link between the Driver for SAP and the SAP System Application Server. Typically, there are several RFC user IDs (one per person).
- **SAP System Administrator** — A SAP System Administrator ID is required for the installation of ABAP programs and function modules, for the configuration of destinations and variant for batch operations, and for setting up authorizations for user IDs to use the Driver for SAP. This user ID is only used for the installation.


### Connectivity

The Driver for SAP and the SAP System Application Server usually use TCP/IP communication. Refer to the RFC documentation from SAP AG. The host of the SAP System

Application Server must be known by the host of the Driver for SAP. Alternatively, you can use the IP address to identify the SAP System Application Server. The TCP/IP services file must contain entries for the services, ports, and protocols used for the communication.

The following is an example for entries in the services file:

```
sapdp00 3200/tcp
sapdp01 3201/tcp
...
sapdp99 3299/tcp
sapgw00 3300/tcp
sapgw01 3301/tcp
...
sapgw99 3399/tcp
sapsp00 3400/tcp
sapsp01 3401/tcp
...
sapsp99 3499/tcp
```

 **Note:** If the SAPGUI is installed on the machine, the TCP/IP services file already contains these entries.

## Authorization Profiles

To install and use the Driver for SAP, a user ID with certain authorizations is required. An authorization has an authorization object. Several authorizations can be bundled together into an authorization profile.

If the batch functionality of the Driver for SAP is used, the RFC user ID needs to have authorization to submit batch jobs already released.

The RFC user IDs require authorizations for the following authorization objects:

Object	Minimum Requirement for Values	Example for Predefined Authorization	Notes
S_RFC (Authorization check for RFC access)	ACTVT: * RFC_NAME: * RFC_TYPE: *	S_RFC_ALL	
S_TABU_DIS (Table maintenance via standard tools such as SM31)	ACTVT: 03 DICBERCLS: *	S_TABU_SHOW	
S_BTCH_JOB (Background processing: Operations on Background Jobs)	JOBACTION: RELE JOBGROUP: *		Only required if batch functionality of the RFC server is used

The existing authorizations, for example S\_TABU\_SHOW, can be used. The S\_RFC and the S\_TABU\_DIS authorizations are in authorization profile A\_ANZEIGE.

# Setting up the Driver for SAP

This section describes the set-up for the Driver for SAP after the software has been installed. Complete the following steps on the Federation Server:

1. [Install the Unicode RFC libraries from SAP](#)
2. [Set the environment variables](#)

## Installing the Unicode RFC Libraries from SAP

The Driver for SAP requires the 64-bit version of the SAP Unicode RFC libraries, Release 7.10. The libraries must be installed on the Federation Server. These shared libraries are provided by SAP AG. Download and install the 64-bit Unicode libraries for your platform, following the instructions in SAP Note 413708.

## Setting the Environment Variables

The Driver for SAP executable uses SAP's shared libraries. You must add the location of the SAP RFC shared libraries to the shared library path environment variable specific to your operating system. For Windows, ensure the shared libraries are installed in the system path, or add the directory of the installed SAP Unicode RFC libraries to the Path environment variable. For UNIX, replace `rfclib_directory` in the table below with the directory where the RFC shared libraries are installed.

AIX	
Bourne Shell	<pre>\$ LIBPATH=rfclib_directory:\$LIBPATH \$ export LIBPATH</pre>
C Shell	<pre>\$ setenv LIBPATH rfclib_directory:\$LIBPATH</pre>

HP-UX	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=rfclib_directory:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH=rfclib_directory:\$LD_LIBRARY_PATH</pre>

HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=rfclib_directory:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH=rfclib_directory:\$LD_LIBRARY_PATH</pre>

Linux for Intel Architecture, Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=rfclib_directory:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH=rfclib_directory:\$LD_LIBRARY_PATH</pre>

# Installing SAP Components

Complete the following steps on the SAP System:

1. [Verify the Prerequisites](#)
2. [Install ABAP Programs and Function Modules](#)
3. [Maintain the RFC Destinations](#)
4. [Maintain the /SAS/DESTS Table for Destination Groups](#)
5. [Activate the BAdI Implementation for SAP BW/BI Authorization Checks](#)

## Verifying the Prerequisites

The following sections identify the SAPGUI and SAP administrator ID prerequisites.

### SAPGUI Prerequisites

The installation of the Driver for SAP components require SAPGUI software to be installed on your PC or workstation.



**Note:** Although it is not absolutely necessary to have the SAPGUI installed on the same PC or workstation where the Driver for SAP is going to be installed, you need access to the SAPGUI during the installation. Because the usage of the SAPGUI complements SAP functionality, it is recommended that the SAPGUI be installed on the same PC or workstation.

### SAP Administrator ID Prerequisites


A valid SAP user ID and password is required. The user must have authorization to transport files and for RFC destination maintenance. It is strongly recommended to get assistance from your SAP System Administrator to perform these tasks.

## Installing ABAP Programs and Function Modules


Delivery transport files are included in the Driver for SAP. These transport files include all of the components, ABAP programs and function modules needed to run the Driver for SAP.

The delivery transports have to be imported on each SAP application server that is going to be accessed by the Federation Server. If an SAP system is upgraded, the delivery transports have to be imported again.


Two sets of transports are included, one for releases prior to SAP Release 7.0 and one for SAP Release 7.0 and above. You must import the transport files that apply to your system.

Version	Transport	Purpose	To be applied to
SAP systems prior to SAP NetWeaver 7.0 (Kernel 6.40 or lower)	SAPKA92220INSAS  <b>Note:</b> This transport	Supports the Driver for SAP	All SAP systems to be accessed by the Driver for SAP



Version	Transport	Purpose	To be applied to
	must be installed first.		
SAP NetWeaver 7.0 based systems and higher	SAPKA92230INSAS  <b>Note:</b> This transport must be installed first.	Supports the Driver for SAP	All SAP systems to be accessed by the Driver for SAP
	SAPKB92331INSAS	Supports new BI 7.0 authorization concept	Optional; SAP BI 7.0 systems and above; only apply if you are using the new authorization concept

To import the transport files to your SAP systems, follow the instructions below. The instructions are based on the usage of the tp program (a utility for transport between SAP systems) on the operating system level.

 **Note:** You should replace *HOME* in these instructions with the actual directory path where the Federation Server is installed.

1. Log in as SAP System Administrator to the SAP application server.
2. Move the transport files from the Federation Server location into the appropriate directories on your SAP systems.
  - For SAP application servers on Windows® systems:  
Copy the `r3trans.exe` file to your SAP application server and extract the files into the transport directory, for example, `HOME:\share\SAP`. The files for all transports will be put into the `cfiles` and `data` subdirectories.
  - For SAP application servers on UNIX® systems:  
Copy the `r3trans.tar` file to your SAP application server and extract the files into the transport directory, for example, `HOME:/share/SAP`. Assuming the tar file is downloaded to the user's `HOME` directory, follow these procedures to extract the files into the `cfiles` and `data` subdirectory in `/usr/sap/trans`.

```
$ cd /usr/sap/trans
$ tar -xvf $HOME/r3trans.tar
```

3. Change to the transport program directory using the following command:

For Windows:

```
<drive>:
cd \usr\sap\trans\bin
```

For UNIX:

```
$ cd /usr/sap/trans/bin
```

4. Load the transport into the transport buffer and import the transport into your SAP system with the following commands. Replace *sid* with the system ID for your SAP system.

```
tp addtobuffer SAPKA92220INSAS sid  
tp import SAPKA92220INSAS sid U2
```

#### **Notes:**

1. Make sure you are using the correct profile for the transport control program tp. In some cases it might be necessary to use the parameter pf= to specify the TPPARAM file.
2. Because the transport file uses a long name, the nbufform=true TP option must be set. The option can either be maintained in the SAP system using transaction STMS, or it can be specified as a parameter to the tp command. Additionally the TP option tp\_version= must be set to at least 264 to allow the long names.
3. The U2 option allows the originals to be overwritten if the user has previously installed these ABAP objects.
4. The transports contain only client-independent ABAP objects. The tp import can therefore use any existing client that is correctly set up for imports. Verify that the ABAP program RDDIMPDP is correctly scheduled in the client you use for the import.
5. If the transport files are imported into a Unicode SAP system, use the transport profile parameter "setunicodeflag=true" to force setting the Unicode flags in the imported programs. Refer to SAP Note 330267 for more details. The "setunicodeflag=true" is not necessary if you are using the transports for SAP NetWeaver 7.0 based system and higher. Those transports have been created with the Unicode flag.

Considering these notes, the tp commands might require additional parameters. Replace *sid* with the system ID for the SAP system.



**Important:** The tp commands listed on several lines in the following examples should be entered on a single command line. Be sure to include a space before adding the text from each of the following lines.

- For SAP Release prior to SAP NetWeaver 7.0 (Kernel 6.40 or lower), non-Unicode SAP Server, on Windows:

```
tp addtobuffer SAPKA92220INSAS sid pf=\usr\sap\trans\bin\TP_DOMAIN_sid.PFL  
-D"nbufform=true" -D"tp_version=264"
```

```
tp import SAPKA92220INSAS sid pf=\usr\sap\trans\bin\TP_DOMAIN_sid.PFL -  
D"nbufform=true" -D"tp_version=264"
```

- For SAP Release prior to SAP NetWeaver 7.0 (Kernel 6.40 or lower), non-Unicode SAP Server, on UNIX:

```
$ tp addtobuffer SAPKA92220INSAS sid  
pf=/usr/sap/trans/bin/TP_DOMAIN_sid.PFL -D"nbufform=true" -  
D"tp_version=264"
```

```
$ tp import SAPKA92220INSAS sid pf=/usr/sap/trans/bin/TP_DOMAIN_sid.PFL -  
D"nbufform=true" -D"tp_version=264"
```

- For SAP Release prior to SAP NetWeaver 7.0 (Kernel 6.40 or lower), Unicode SAP Server, on Windows:  

```
tp addtobuffer SAPKA92220INSAS sid pf=\usr\sap\trans\bin\TP_DOMAIN_sid.PFL -D"nbufform=true" -D"tp_version=264" -D"setunicodeflag=true"

tp import SAPKA92220INSAS sid pf=\usr\sap\trans\bin\TP_DOMAIN_sid.PFL -D"nbufform=true" -D"tp_version=264" -D"setunicodeflag=true"
```
- For SAP Release prior to SAP NetWeaver 7.0 (Kernel 6.40 or lower), Unicode SAP Server, on UNIX:  

```
$ tp addtobuffer SAPKA92220INSAS sid pf=/usr/sap/trans/bin/TP_DOMAIN_sid.PFL -D"nbufform=true" -D"tp_version=264" -D"setunicodeflag=true"

$ tp import SAPKA92220INSAS sid pf=/usr/sap/trans/bin/TP_DOMAIN_sid.PFL -D"nbufform=true" -D"tp_version=264" -D"setunicodeflag=true"
```
- For SAP NetWeaver 7.0 based systems and higher, Unicode SAP Server, on Windows:  

```
tp addtobuffer SAPKA92230INSAS sid pf=\usr\sap\trans\bin\TP_DOMAIN_sid.PFL -D"nbufform=true"

tp import SAPKA92230INSAS sid pf=\usr\sap\trans\bin\TP_DOMAIN_sid.PFL
```
- For SAP NetWeaver 7.0 based systems and higher, Unicode SAP Server, on UNIX:  

```
$ tp addtobuffer SAPKA92230INSAS sid pf=/usr/sap/trans/bin/TP_DOMAIN_sid.PFL -D"nbufform=true"

$ tp import SAPKA92230INSAS sid pf=/usr/sap/trans/bin/TP_DOMAIN_sid.PFL -D"nbufform=true"
```

## Maintaining RFC Destinations



**Note:** If the Driver for SAP will execute requests using the SAP batch processing facility (recommended), you must complete this section.

The Driver for SAP uses multiple RFC destinations (TCP/IP connection type) for accessing a SAP System in batch. The number of destinations setup for the Driver for SAP limits the number of concurrent requests to the SAP application server.

For example, create 6 destinations with connection type T and activation type Registered Server Program that can be used by the SAS® server. The program ID for the registered server program must be unique on the SAP gateway.

RFC Destination Name	Program ID
SAS1	RFC.SAS1
SAS2	RFC.SAS2
SAS3	RFC.SAS3
SAS4	RFC.SAS4
SAS5	RFC.SAS5
SAS6	RFC.SAS6

Complete the following steps:

1. Call transaction SM59 in SAP. Specify transaction code **/nsm59** in the command field.
2. Click **Create**.
3. Enter **SAS1** as the RFC destination.
4. Enter **T** as the Connection type.
5. Enter a description for the destination.
6. Click **Enter**.
7. Choose Registration for the Activation Type or Registered Server Program in the Technical Settings tab.
8. Enter the RFC.SAS1 as the program ID.
9. If required, enter the gateway host and gateway service in the Gateway Options panel. The gateway host is the host name of the local gateway and gateway service is usually *sapgwsysnr*, where *sysnr* is replaced by the system number of the SAP system.
10. For Unicode SAP systems, select the Unicode on the MDMP & Unicode tab. Ignore the message about performing the Unicode test. The Unicode test cannot be performed with the destinations created for the Driver for SAP.
11. Save the destination.
12. Repeat the steps for each of the new RFC destinations.

## Maintaining the /SAS/DESTS Table

The RFC destinations defined in the previous step must be grouped into destination groups. The groups are defined in table /SAS/DESTS which is used for controlling the access to the destinations from all Federation Servers that access the SAP system.

The destination group is a parameter of the Driver for SAP. The default is "SAS1".

Complete the following steps:

1. Call transaction SM30 in SAP. In the command line, type transaction code **/nsm30**.
2. In the **Table** field, type the table name **/SAS/DESTS**.
3. In the **Restrict Data Range** field, select **No Restrictions**.
4. Click **Maintain**.
5. In the information message dialog, click **OK**.
6. Click **New Entries**.
7. For each of the RFC destinations that you defined in step 2, enter the destination group ID as the SAS ID and the RFC destination name. For example, to define the destinations for destination group SAS1:

SAS ID	RFC Destination	Used
SAS1	SAS1	
SAS1	SAS2	
SAS1	SAS3	
SAS1	SAS4	
SAS1	SAS5	
SAS1	SAS6	

8. Save the table.

## Activating BAdI Implementation

The Driver for SAP has three basic implementations for table access authorization checks. The default implementation uses the SAP authorization object S\_TABU\_DIS to check the authorization. If you want to use any of the other two implementations you have to activate the appropriate BAdI implementation.

BAdI Implementation	
<i>Default</i>	Authorization object S_TABU_DIS
Classic BAdI /SAS/AUTHBW01	For BW and BI: User authorization checks at the InfoCube, InfoObject and ODS level using the reporting authorization (SAP standard authorization concept).
New BAdI enhancement /SAS/IM_AUTHBI01	For BI 7.0+ only: User authorization checks using the analysis authorization. This not only provides a authorization check for the infoProvider (infoCube, infoObject, DSO) but also column level restrictions on master data attributes and key figures, and row level restrictions on attributes.

In releases prior to SAP NW BI 7.0 SAP uses the reporting authorization concept that uses the SAP standard authorization concept. If you want to activate the SAS implementation for those authorization checks, follow these steps:

1. Call transaction SE19 in SAP. In the command field, type transaction code **/nse19**.
2. Enter **/SAS/AUTHBW01** as the implementation.
3. Click **Activate**.

In BI 7.0 SAP introduces a new authorization concept for analysis authorization. If you want to use the SAS implementation for those authorization checks import the appropriate transport (SAPKB92331INSAS). The implementation is activated by default. If you want to deactivate the implementation, follow these steps:

1. Call transaction SE19 in SAP. In the command field, type transaction code **/nse19**.
2. In the Edit Implementation field, select the **New BAdI** check box.
3. Enter **/SAS/IM\_AUTHBI01** as the enhancement implementation.
4. Click **Change**.

5. Double-click the BAdI Implementation to deactivate (such as /SAS/BADI\_CHECK\_FILTER) and unselect the Implementation is active check box in the Runtime Behavior field. Repeat for each of the implementations listed in the left hand side of the **Enh. Implementation Elements** tab.
6. Save and activate the changes.

# Configuring the UNIX Environment

After installing DataFlux® Federation Server (Federation Server) in a UNIX® environment, additional configuration is needed.

- [Set the LANG Environment Variable](#)
- [Configuring the Federation Server Drivers](#)

## Set the LANG Environment Variable

If using BASE data sets with the Federation Server, the LANG environment variable must be set before bringing up the Federation Server. This environment variable is primarily needed for the VALIEDATEFMT table.

The following is for UNIX only; it does not apply to Windows environments.

Most UNIX or Linux systems use the LANG variable to specify the desired locale and this variable is often already set in your environment. Locale names vary among different UNIX or Linux operating systems so use a LANG value that is supported by your version of UNIX or Linux.

Invoke the **locale** command to show your current locale. Use **locale -a** to display a list of all the locales that are currently installed on the machine.

For more information on setting locale environment variables, consult the documentation for your operating system.

## Configuring the Federation Server Drivers

To determine if the combination of your operating system and the version or release of your database management system (DBMS) is supported by the Federation Server driver you want to configure, see [System Requirements](#).

Before configuring the Federation Server driver software, you should determine the following information about your DBMS:

- The version or release of the DBMS client shared libraries installed on your operating system. This is important due to potential incompatibilities between DBMS versions or releases.
- The location of the DBMS client shared libraries. This is important so that Federation Server driver software can be loaded at execution time.



**Note:** The steps outlined in this chapter assume that the Federation Server drivers have been installed.

Refer to the following sections for detailed DBMS-specific instructions on configuring your environment to interface with your Federation Server driver software:

- [Driver for SAP](#)

- [Driver for DB2](#)
- [Driver for MySQL](#)
- [Driver for ODBC](#)
- [Driver for Oracle](#)
- [Driver for Teradata](#)

## Driver for SAP

SAP® software requires extensive configuration before it can be used. For more information, see [Installing and Configuring the Driver for SAP](#).

## Driver for DB2

The DataFlux Driver for DB2 uses shared libraries, referred to in UNIX as shared objects. You must add the location of the shared libraries to one of the system environment variables, and, if necessary, indicate the DB2® version that you have installed at your site. Before setting the environment variables as shown in the examples below, you must also set the following environment variables:

- The INSTHOME environment variable must be set to your DB2 home directory.
- The DB2DIR environment variable should also be set to the value of INSTHOME.
- The DB2INSTANCE environment variable should be set to the DB2 instance configured by the administrator.

AIX	
Bourne Shell	<code>\$ LIBPATH=\$INSTHOME/lib:\$LIBPATH</code> <code>\$ export LIBPATH</code>
C Shell	<code>\$ setenv LIBPATH \$INSTHOME/lib:\$LIBPATH</code>

HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	<code>\$ SHLIB_PATH=\$INSTHOME/lib:\$SHLIB_PATH</code> <code>\$ export SHLIB_PATH</code>
C Shell	<code>\$ setenv SHLIB_PATH \$INSTHOME/lib:\$SHLIB_PATH</code>

Linux for Intel Architecture, Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	<code>\$ LD_LIBRARY_PATH=\$INSTHOME/lib:\$LD_LIBRARY_PATH</code> <code>\$ export LD_LIBRARY_PATH</code>
C Shell	<code>\$ setenv LD_LIBRARY_PATH \$INSTHOME/lib:\$LD_LIBRARY_PATH</code>

## Driver for MySQL

The DataFlux Driver for MySQL uses shared libraries, referred to in UNIX as shared objects. You must add the location of the MySQL shared libraries to the shared library path environment variable specific to your operating system.



## Setting the Shared Libraries

Due to the way MySQL Client libraries are distributed on most operating systems, you will need to link the Driver for MySQL module against the MySQL client libraries on your system using the following process:

1. Set the following environment variables, where `MYSQL_CLIENT_DIR=` is the directory location for the newly-linked MySQL client module. It needs to be set on your `PATH` so the module can be loaded (variable not needed on Linux for Intel Architecture and Linux for Itanium-based Systems). `MYSQL_LIBDIR` is the MySQL client installation location. Use the location of the dynamic client libraries. By default, the location is `/usr/lib` on Linux for Intel Architecture and Linux for Itanium-based systems.

For example, if MySQL is installed in the `/usr/local/mysql/5.0` directory, do the following:

<b>AIX, HP-UX, HP-UX for the Itanium Processor Family Architecture, Solaris, and Solaris for x64</b>	
Bourne Shell	<pre>\$ MYSQL_CLIENT_DIR= install_directory/dfs/lib \$ export MYSQL_CLIENT_DIR \$ MYSQL_LIBDIR=/usr/local/mysql/5.0/lib \$ export MYSQL_LIBDIR</pre>
C Shell	<pre>\$ setenv MYSQL_CLIENT_DIR install_directory/dfs/lib \$ setenv MYSQL_LIBDIR /usr/local/mysql/5.0/lib</pre>

<b>Linux for Intel Architecture and Linux for x64</b>	
Bourne Shell	<pre>\$ MYSQL_LIBDIR=/usr/lib \$ export MYSQL_LIBDIR</pre>
C Shell	<pre>\$ setenv MYSQL_LIBDIR /usr/lib</pre>

2. Modify the shared library variable based on the host and shell you are using, according to the following tables:

<b>AIX</b>	
Bourne Shell	<pre>\$ LIBPATH=\$MYSQL_CLIENT_DIR:\$LIBPATH \$ export LIBPATH</pre>
C Shell	<pre>\$ setenv LIBPATH \$MYSQL_CLIENT_DIR:\$LIBPATH</pre>

<b>HP-UX and HP-UX for the Itanium Processor Family Architecture</b>	
Bourne Shell	<pre>\$ SHLIB_PATH=\$MYSQL_CLIENT_DIR:\$SHLIB_PATH \$ export SHLIB_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH \$MYSQL_LIBDIR:\$LD_LIBRARY_PATH</pre>

<b>Solaris and Solaris for x64</b>	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=\$MYSQL_CLIENT_DIR:\$MYSQL_LIBDIR:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH \$MYSQL_CLIENT_DIR:\$MYSQL_LIBDIR:\$LD_LIBRARY_PATH</pre>

Linux for Intel Architecture and Linux for x64	
Bourne Shell	<code>\$ LD_LIBRARY_PATH=\$MYSQL_LIBDIR:\$LD_LIBRARY_PATH</code> <code>\$ export LD_LIBRARY_PATH</code>
C Shell	<code>\$ setenv LD_LIBRARY_PATH \$MYSQL_LIBDIR:\$LD_LIBRARY_PATH</code>

- Once the environment variables are set, run the link command, as necessary, based on your operating system. This command will produce the **libmysqlclient\_os** file in the directory that was set as the value for **MYSQL\_CLIENT\_DIR**. You must have write permissions to this directory.

## AIX

```
ld -berok -b64 -bM:SRE -bexpall -e _nostart
-o $MYSQL_CLIENT_DIR/libmysqlclient_aix install_directory/dfs/lib/r64myl.o
-lpthreads -L$MYSQL_LIBDIR -lmysqlclient -lm -lc
```

## Linux for Itanium-based Systems

A link is not required for Linux for Itanium-based Systems because it supports the MySQL dynamic library.

## Linux for Intel Architecture

A link is not required for Linux for Intel Architecture because it supports the MySQL dynamic library.

## HP-UX

```
ld -b -o $MYSQL_CLIENT_DIR/libmysqlclient_hp
install_directory/dfs/lib/h64myl.o
-L$MYSQL_LIBDIR -lmysqlclient -lc -lm
```

## HP-UX for the Itanium Processor Family

```
ld -b -o $MYSQL_CLIENT_DIR/libmysqlclient_hpi
install_directory/dfs/lib/h6imyl.o
-L$MYSQL_LIBDIR -lmysqlclient -lc -lm -lz
```

## Solaris and Solaris for x64

Include the location of the linker in the **LD\_LIBRARY\_PATH** environment variable, which is usually found in the libraries listed in the example below:

Bourne Shell	<code>\$ LD_LIBRARY_PATH=/usr/lib/???:/usr/ucblib/???:\$LD_LIBRARY_PATH</code> <code>\$ export LD_LIBRARY_PATH</code>
C Shell	<code>\$ setenv LD_LIBRARY_PATH</code> <code>/usr/lib/???:/usr/ucblib/???:\$LD_LIBRARY_PATH</code>

Fill the **???** variable based on the hardware of the Solaris system:

- SPARC: **sparcv9**
- Solaris for x64 on an Intel system: **64**
- Solaris on x64 on an AMD system: **amd64**

Then use the link command:

For Solaris:

```
ld -64 -G -o $MYSQL_CLIENT_DIR/libmysqlclient_sun
install_directory/dfs/lib/s64myl.o -L$MYSQL_LIBDIR -lmysqlclient
```

For Solaris on x64:

```
ld -64 -G -o $MYSQL_CLIENT_DIR/libmysqlclient_sax
install_directory/dfs/lib/saxmyl.o -L$MYSQL_LIBDIR -lmysqlclient
```

## Driver for ODBC

You may have to edit the `.odbc.ini` file in your home directory with a text editor to configure data sources. Some DataFlux Driver for ODBC (Driver for ODBC) vendors may allow system administrators to maintain a centralized copy by setting the environment variable `ODBCINI`. Please refer to your Driver for ODBC vendor documentation to find more specific information.

The Drivers for ODBC are ODBC API-compliant shared libraries, referred to in UNIX as shared objects. You must add the location of the shared libraries to one of the system environment variables so that Drivers for ODBC can be loaded dynamically at run time. You must also set the `ODBCHOME` environment variable to your ODBC home directory before setting the environment variables as shown in the examples.

Linux for Intel Architecture and Linux for x64	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=\$ODBCHOME/lib:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH \$ODBCHOME/lib:\$LD_LIBRARY_PATH</pre>

Solaris and Solaris for x64	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=\$ODBCHOME/lib:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH \$ODBCHOME/lib:\${LD_LIBRARY_PATH}</pre>

AIX	
Bourne Shell	<pre>\$ LIBPATH=\$ODBCHOME/lib:\$LIBPATH \$ export LIBPATH</pre>
C Shell	<pre>\$ setenv LIBPATH \$ODBCHOME/lib:\${LIBPATH}</pre>

HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	<pre>\$ SHLIB_PATH=\$ODBCHOME/lib:\$SHLIB_PATH \$ export SHLIB_PATH</pre>
C Shell	<pre>\$ setenv SHLIB_PATH \$ODBCHOME/lib:\${SHLIB_PATH}</pre>

## Driver for Oracle

### Connecting to an Oracle database

If you plan to connect to an Oracle® 10 or Oracle 11 database using a Federation Server on UNIX, you may need to copy files needed for Oracle. By default, Oracle 10 is used.

#### Oracle 10

For Oracle 10, run the following commands:

```
cd /installation_directory/dfs/lib
mv tkeora10.so tkeora.so
```

#### Oracle 11

For Oracle 11, run the following commands:

```
cd /installation_directory/dfs/lib
mv tkeora11.so tkeora.so
```



**Note:** For HP64, the file extensions are .sl.

### Setting the ORACLE\_HOME Variable

In order to use DataFlux Driver for Oracle (Driver for Oracle), you must set the ORACLE\_HOME environment variable. In addition, you must make sure that the shared library path variable (the name of this variable is operating system dependent) points to where the Oracle shared libraries are located. This is required since the Driver for Oracle executable uses Oracle shared libraries and needs to know where they are located at your site.

The following are examples for the various operating systems:

AIX	
Bourne Shell	<pre>\$ LIBPATH=\$ORACLE_HOME/lib:\$LIBPATH \$ export LIBPATH</pre>
C Shell	<pre>\$ setenv LIBPATH=\$ORACLE_HOME/lib:\$LIBPATH</pre>

HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	<pre>\$ SHLIB_PATH=\$ORACLE_HOME/lib:\$SHLIB_PATH \$ export SHLIB_PATH</pre>
C Shell	<pre>\$ setenv SHLIB_PATH \$ORACLE_HOME/lib:\$SHLIB_PATH</pre>

Linux for Intel Architecture, Linux for Itanium-based Systems, Solaris, and Solaris for x64	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=\$ORACLE_HOME/lib:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH \$ORACLE_HOME/lib:\$LD_LIBRARY_PATH</pre>

# Driver for Teradata

## Access to Shared Libraries

The DataFlux Driver for Teradata uses shared libraries, referred to in UNIX as shared objects. These shared objects typically reside in `/usr/lib`. You must add the location of the shared libraries to one of the system environment variables.

AIX	
Bourne Shell	<pre>\$ LIBPATH=TERADATA-CLIENT-LOCATION:\$LIBPATH \$ export LIBPATH</pre>
C Shell	<pre>\$ setenv LIBPATH TERADATA-CLIENT-LOCATION:\$LIBPATH</pre>

HP-UX	
Bourne Shell	<pre>\$ SHLIB_PATH=TERADATA-CLIENT-LOCATION:\$SHLIB_PATH \$ export SHLIB_PATH</pre>
C Shell	<pre>\$ setenv SHLIB_PATH TERADATA-CLIENT-LOCATION:\$SHLIB_PATH</pre>

HP-UX for the Itanium Processor Family	
Bourne Shell	<pre>\$ SHLIB_PATH=TERADATA-CLIENT-LOCATION:\$SHLIB_PATH \$ export SHLIB_PATH \$ LD_PRELOAD=/usr/lib/hpux64/libpthread.so.1 \$ export LD_PRELOAD</pre>
C Shell	<pre>\$ setenv SHLIB_PATH TERADATA-CLIENT-LOCATION:\$SHLIB_PATH \$ setenv LD_PRELOAD /usr/lib/hpux64/libpthread.so.1</pre>

Linux for Intel Architecture, Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	<pre>\$ LD_LIBRARY_PATH=TERADATA-CLIENT-LOCATION:\$LD_LIBRARY_PATH \$ export LD_LIBRARY_PATH</pre>
C Shell	<pre>\$ setenv LD_LIBRARY_PATH TERADATA-CLIENT- LOCATION:\$LD_LIBRARY_PATH</pre>

## TTU 8.2 and HP-UX

HP-UX users with TTU 8.2 must create two symbolic links from the `/usr/lib/pa20_64` directory with the following commands:

```
$ ln -s /usr/lib/pa20_64/libicudatatd.sl libicudatatd.sl.34
$ ln -s /usr/lib/pa20_64/libicuuctd.sl libicuuctd.sl.34
```

# Configuring DataFlux Federation Server

This section covers configuring DataFlux® Federation Server (Federation Server) for Microsoft® Windows® and UNIX® operating systems.

The following topics cover how to configure server software:

- [Configuring the DataFlux Federation Server Service](#)
- [Configuring Connections](#)
- [Configuring Licensing](#)
- [Users, Groups, and Roles](#)
- [Managing Configuration Files and Scripts](#)

## Configuring the Federation Server Service

The Federation Server runs as a Microsoft Windows service. You can start and stop the service using the Microsoft Management Console (MMC).

In UNIX, Federation Server runs as a daemon administered from a command line. The dfsadmin application is used to start and stop the daemon.

## Starting and Stopping Federation Server in Windows

When installed in a Windows environment, Federation Server runs as a Windows service named DataFlux Federation 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 Federation Server**.
6. Click either **Stop the service** or **Restart the service**.

## Modifying Federation Server Windows Service Log On

When Federation Server is installed, it creates a service named DataFlux Federation 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 Federation Server log on:

1. Select **Control Panel > Administrative Tools**.
2. Double-click **Services**, and select the **DataFlux Federation 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 Federation Server Daemon in UNIX

Start and stop the daemon using the dfsadmin application included in the installation. This application can be run using the command-line command: `./bin/dfsadmin yourcommand` from the installation root directory.

The value of *yourcommand* should be one of the following:

Command	Description
start	Starts the Federation Server.
stop	Stops the Federation Server.
status	Checks whether or not the Federation Server is running.
help	Displays the command usage and options.
version	Displays the version information.

For example, to start the server:

```
./bin/dfsadmin start
```

To stop the server:

```
./bin/dfsadmin stop
```

## Configuring Connections

This section describes how to configure Federation Server connections in Windows and UNIX environments.

To access a database through ODBC with the Federation Server, a Driver for ODBC for the specific database must be used. The Federation Server comes with the Drivers for ODBC for several databases. The database must also be configured as an ODBC data source.


To access a database through a vendor supplied client, the client must be installed and configured according to the vendor documentation.

## Windows

To add an ODBC data source, use the ODBC Data Source Administrator provided with Microsoft Windows.

To set up a new ODBC connection:

1. Click **Start > Control Panel**.
2. Double-click **Administrative Tools > Data Sources (ODBC)**,

 **Note:** In Windows 7, the view of the Control Panel can vary. If you don't see Administrative Tools when you open the Control Panel, click **System and Security** to access **Administrative Tools, Data Sources (ODBC)**.

1. Click **Add**.
2. In the **ODBC Data Source Administrator** dialog, select the driver that is appropriate for your data source.
3. In the **ODBC Driver Setup** dialog, enter the **Data Source Name, Description**, and **database-specific configuration options**. These values are required, and can be obtained from your database administrator.

For non-ODBC connections, use the vendor supplied client configuration utility.

## UNIX

Use the interactive ODBC Configuration Tool, `dfdbconf`, to add new data sources to the ODBC configuration.

1. From the root directory of the Federation Server installation, run: `./bin/dfdbconf`.
2. Select **A** to add a data source. You can also use `dfdbconf` to delete a data source if it is no longer needed.
3. Select a template for the new data source by choosing a number from the list of available drivers.
4. You are prompted to set the appropriate parameters for that driver. The new data source is then added to your `odbc.ini` file.

Once you have added all of your data sources, the interactive ODBC Viewer application, `dfdbview`, can be used to test your connection. For example, if you added a data source called `my_oracle`, run: `./bin/dfdbview my_oracle` (from the installation root) to test the connection. You may be prompted for a user name and password. If the connection succeeds, you will see a prompt from which you can enter SQL commands and query the database. If the connection fails, Federation Server displays error messages describing one or more reasons for the failure.

For non-ODBC connections, use the vendor supplied client configuration utility. For more information on configuring third-party databases, see [Configuring the DataFlux Federation Server Drivers](#).



# Configuring Licensing

Federation 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 Federation Server has uncounted licenses (an infinite number of licenses) for each type of license purchased. If Federation Server is packaged as part of SAS®, you have the option of selecting SAS license file as your licensing method.

The following sections provide information on how to configure DataFlux licenses. For SAS licenses, contact your sales executive to receive the license. Then, complete step 4, depending on your operating system.

## Windows

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

1. To display the DataFlux Host ID, click **Start > Programs > DataFlux > Show Host ID**. Then, click **Copy to Clipboard** to copy the ID number.
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, and then click **Submit**.
4. When you receive your new license file, save it to the `drive:\Program Files\DataFlux\FederationServer\version\etc\license\`. License files must have a `.lic` file name extension in order to be considered.

The location of the license file is specified in the `dfs_serv.xml` file using the **License Option Set**. Here, you can indicate the primary and secondary license types and update the location of these license files. For more information on the options in the `dfs_serv.xml` file, see [Federation Server Configuration File](#).

## UNIX

To configure your license file for Federation Server in UNIX, 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, and then click **Submit**.
4. When you receive your new license file, save it on the UNIX server in the `/opt/dataflux/dfs/etc/license` directory. License files must have a `.lic` file name extension in order to be considered.

## Annual Licensing Notification

For DataFlux licenses, thirty days prior to license expiration, you will receive a message that your license will expire in a certain number of days. If you have a SAS license (setinit), this message is defined by the warning period. This is configurable through SAS.



**Note:** DataFlux licenses are not configurable.

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

## Managing Configuration Files and Scripts

Configuration files and scripts control the way that the Federation Server operates. These files are stored in the Federation Server installation folder. For information about the initial installation and configuration for the Federation Server, see [Installing DataFlux Federation Server](#).

The following table lists the configuration files and scripts:

File or Script Name	Description
dfs_serv.xml	<p>Specifies the SYSTEM users, the location of the internal database, and other Federation Server configuration settings. This file is installed in the <b>\etc</b> directory of the installation path.</p> <p>For more information, see <a href="#">Federation Server Configuration File</a>.</p>
dfs_log.xml	<p>Specifies the logging facility configuration for the Federation Server. This file is installed in the <b>\etc</b> directory of the installation path.</p> <p>For more information, see <a href="#">Logging Facility Configuration File</a>.</p>

## Federation Server Configuration File

The following options are available in the **dfs\_serv.xml** configuration file for the Federation Server:

### AppendEnv

#### Syntax

```
<OptionSet name="AppendEnv">  
  <Option name="FIREBIRD">drive:\install_loc\firebird</Option>  
</OptionSet>
```

#### Description

The AppendEnv option locates the specified OS environment variable and appends the specified option to the environment variable's current value. If the environment variable does not exist, it is created and set to the specified value. The AppendEnv option does not

add a delimiter between the existing and appended environment variable values; therefore, if a delimiter is needed, it should be included at the beginning of the specified value.

Each AppendEnv, PrependEnv and SetEnv option is processed entirely in the order that they appear in the configuration file.

## SetEnv OptionSet

### Syntax

```
<OptionSet name="SetEnv">  
  <Option name="FIREBIRD">drive:\install_dir\lib\fbembed</Option>  
  <Option name="FIREBIRD_LOG">drive:\install_dir\var\log</Option>  
</OptionSet>
```

### Description

The SetEnv option sets OS environment variables to specific values. If the environment variable does not exist, it will be created and set to the option value. If the environment variable does exist, the value will be updated to the option value.

Each AppendEnv, PrependEnv, and SetEnv option is processed entirely in the order that they appear in the Configuration file.

## Transactional Data Store Options

### Syntax


```
<Option name="FIREBIRD">drive:\install_dir\lib\fbembed</Option>  
<Option name="FIREBIRD_LOG">drive:\install_dir\var\log</Option>
```

### Description

The FIREBIRD environment variable specifies the location of the Transactional Data Store installation files.

The FIREBIRD\_LOG environment variable specifies the location of the log files for Transactional Data Store. The configuration file generated during installation sets the FIREBIRD\_LOG option to the var\log directory of the installation path. If FIREBIRD\_LOG is not set, the federation server will default to one of two locations:

1. TranPath: If the TranPath environment variable is set, FIREBIRD\_LOG is set to the TranPath value.
2. ContentRoot: If TranPath is not set, FIREBIRD\_LOG is set to the ContentRoot value as defined in the configuration file.

 **Note:** For UNIX installations, the dfsadmin script sets the FIREBIRD\_LOG option to <install\_root>/var/log when it is executed. If FIREBIRD\_LOG is set in the environment, it is overwritten by the script.

## TranPath

### Syntax

```
<Option name="TranPath">directory</Option>
```

## Description

Identifies the location in which to store the Transactional Data Store files.

By default, the Transactional Data Store files are stored in ContentRoot as defined in the configuration file. Transactional Data Store system files cannot be stored on remote file systems. If ContentRoot is a network file system or share, use the TranPath option to redirect the Transactional Data Store files to a local directory on the machine where Federation Server is installed.

## Rules and Dependencies

Following are the rules and dependencies for the TranPath option:

1. A relative directory specified in the ContentRoot tag is resolved against the server's working directory.
2. A relative directory specified in the TranPath tag is resolved against the directory specified in the ContentRoot tag, or the working directory of the server if none is specified.
3. If the TranPath option is omitted, it defaults to the directory specified in the ContentRoot tag, or the working directory of the server if none is specified.
4. The TranPath directory is used to resolve the server's system catalog database name specified in the provider specific SecurityProvider/Database tag for the tkescfb provider.

## PrependEnv

### Syntax

```
<OptionSet name="PrependEnv">  
  <Option name="FIREBIRD">drive:\install_loc\firebird</Option>  
</OptionSet>
```

### Description

The PrependEnv option will find the indicated OS environment variable and prepend the option value to the OS environment variable value. If the environment variable did not exist, it will be created and set to the option value. The PrependEnv option will not add a delimiter of any sort between the existing and new environment variable value. If a semicolon (;) is needed, then the option value should include it at the end.

Each AppendEnv, PrependEnv and SetEnv option is processed entirely in the order that they appear in the configuration file.

## SystemUsers

### Syntax

```
<SystemUsers>  
  <Option name="Account">domain\uid1</Option>  
  <Option name="Account">domain\uid2</Option>  
</SystemUsers>
```

## Description

The System Users option defines the administrator user accounts for the Federation Server. These accounts are given *administer* privilege to the Federation Server and all associated data services. This privilege cannot be revoked or denied. When system users grant or deny privileges to others, the grantor is reflected in the system tables as the SYSTEM userid. Each system user should be a domain-qualified user name.

## SecurityProvider OptionSet

### Syntax

```
<OptionSet name="SecurityProvider">
  <Option name="extension">extension_name</Option>
  <Option name="Database">database_path</Option>
</OptionSet>
```

### Description

The SecurityProvider OptionSet provides information about the Federation Server security provider, including the TK extension name and other information specific to the security provider.

## Extension Option

### Syntax

```
<Option name="extension">extension_name</Option>
```

### Description

Identifies the TK extension of the security provider. This value is **tkescfb** for the Transactional Data Store security provider.

## Database Option

### Syntax

```
<Option name="Database">database_path</Option>
```

### Description

Identifies the path to the database (syscat.tdb) to be used for the Transactional Data Store security provider (tkescfb).

The Windows default database path is *drive:\Program Files\DataFlux\FederationServer\version\var\syscat.tdb*. The default database path for UNIX is *<install\_dir>/dfs/var/syscat.tdb*.



**Important:** The **syscat** database must reside on a local file system as indicated by the default installation paths above. If the **var** directory is pointed to a remote file system such as a network file share or storage area network (SAN), the syscat database cannot be created, resulting in multiple application errors.

## ContentRoot Option

### Syntax

```
<Option name="ContentRoot">content_root_path</Option>
```

### Description

Defines the content root for the Federation Server. The content root is used to resolve all relative path names specified in Federation Server configuration, such as a schema path. The content root value is absolute or relative to the installation directory.

### Rules

If the ContentRoot option is not set, files will be written to the install directory.

- Content root is absolute or relative to the install directory.
- TRACEFILEPATH is absolute or relative to content root.
- TRACEFILE paths are resolved against TRACEFILEPATH path. Paths that don't match should be rejected.
- PRIMARYPATH paths in schema configuration options are absolute or relative to content root.
- SCHEMA=(PRIMARYPATH) connection string options are resolved against PRIMARYPATH schema configuration path.

## Port Option

### Syntax

```
<Option name="Port">port_number</Option>
```

### Description

Indicates the port on which the Federation Server will start.

## Authentication Server Option

### Syntax

```
<OptionSet name="AuthenticationServer">  
  <OptionSet name="PrimaryServer">  
    <Option name="URI">IOM-URI</Option>  
  </OptionSet>  
</OptionSet>
```

### Description

Defines the location of the Authentication Server that serves as the backend for the Federation Server.

IOM\_URI contains connection information for the Authentication Server and uses the following format:

```
iom://<machine>:<port>;Bridge;CLSID=2D1BCDBF-F900-4CA9-85F6-95ECDBAF2122
```

For example, IOM\_URI can be used to connect to an Authentication Server that is running on port number 21030 and machine name myhost. The CLSID for an Authentication Server should always be 2D1BCDBF-F900-4CA9-85F6-95ECDBAF2122:

```
iom://yourserver.yourdomain.com:21030;Bridge;CLSID=2D1BCDBF-F900-4CA9-85F6-95ECDBAF2122
```

## License OptionSet

### Syntax

```
<OptionSet name="License">
  <OptionSet name="Primary">
    <Option name="Provider">SAS|DATAFLUX</Option>
    <Option name="Location">path_to_license_file</Option>
  </OptionSet>
  <OptionSet name="Secondary">
    <Option name="Provider">SAS|DATAFLUX</Option>
    <Option name="Location">path_to_license_file</Option>
  </OptionSet>
</OptionSet>
```

### Description

Provides information about the type of licensing performed by the Federation Server. Choices include SAS and DATAFLUX. Both license methods may be enabled, one will be identified as the primary license provider, while the other will be the secondary license provider.

## Primary OptionSet

### Syntax

```
<OptionSet name="Primary">
  <Option ...>
</OptionSet>
```

### Description

Specifies the primary license file.

## Secondary OptionSet

### Syntax

```
<OptionSet name="Secondary">
  <Option ...>
</OptionSet>
```

### Description

Specifies the secondary license file, if applicable.

## Primary and Secondary Options

### Syntax

```
<Option name="Provider"> SAS | DATAFLUX</Option>
```

## Description

Provides information about the license provider. Possible license options are SAS and DATAFLUX.

## Location Option

### Syntax

```
<Option name="Location">
```

## Description

Provides information about the location of the file or server required by the license provider. For SAS, the location option will point to a **setinit** file. The DATAFLUX location option will point to a license server or license file.

## ObjectServerParms Option

### Syntax

```
<Option name="ObjectServerParms">object_server_parameters</Option>
```

## Description

Identifies IOM object server parameters. Values for *object\_server\_parameters* include:

- Clientencryptionlevel = (EVERYTHING | CREDENTIALS | NONE) — Specifies the client encryption level to use. Valid values include:
  - **NONE** — Nothing is encrypted.
  - **CREDENTIALS** — Login credentials are encrypted. Note: these are the login credentials used to authenticate to the Federation Server, and NOT logins used as outbound credentials to connect to third party databases. It also does NOT include credentials passed in administration DDL, such as CREATE ACCOUNT REGISTRATION or CREATE USER.
  - **EVERYTHING** — All client-server communications are encrypted. Setting a value of EVERYTHING may affect server performance.

## NetworkEncryptAlgorithm Option

### Syntax

```
<Option name="NetworkEncryptAlgorithm">algorithm | ("algorithm1", "algorithm2", ...)  
</Option>
```

## Description

- **algorithm | ("algorithm1", "algorithm2", ...)** — Specifies the algorithm or algorithms that can be used for encrypting data that is transferred between a client and a server across a network. When you specify two or more encryption algorithms, use a space or a comma to separate them, and enclose the algorithms in parentheses. If more than one algorithm is specified, the client session negotiates the first specified



algorithm with the server session. If the client session does not support that algorithm, the second algorithm is negotiated, and so on.

This is set with the CLIENTENCRYPTIONLEVEL OBJECTSERVERPARMS option.

Possible values include: SASProprietary and AES.

For more information on Advanced Encryption Standard (AES), see the *DataFlux Authentication Server Administrator's Guide*.

## Logging Facility Configuration File

The logging facility is a flexible, configurable framework you use for collecting, categorizing, and filtering events that are generated by DataFlux processes and writing events to a variety of output devices. The **dfs\_log.xml** configuration file controls the destination, contents, and formats of the logging facility log for the Federation Server. The configuration file specifies options for loggers, appenders, levels, filters, and the layout of log messages.

DataFlux supplies a default XML configuration file, which you can customize. For information, see [Logging Facility](#).

# DataFlux Federation Server Concepts

Managing the DataFlux® Federation Server (Federation Server) requires understanding the following key concepts:

- [Data Services](#)
- [Data Source Names \(DSNs\)](#)
- [Users, Groups and Roles](#)
- [Authorization and Security](#)
- [Database Functionality Supported by Drivers](#)

For additional information about the key concepts, including SYSTEM users, administrators, and process users, see [Best Practices](#).

## Data Services

To access data, the Federation Server administrator must create and configure data services. Data services contain information about how to connect to a particular source of data, such as Oracle® or Base SAS® data sets. Data services allow for the specification of a Federation Server driver. Most data services may be accessed through either a Federation Server driver or a Driver for ODBC.

Data services allow a logical catalog name to be applied to the data source when that data source does not support native catalogs. For data services that do not support catalogs, the Federation Server enables you to define a logical catalog name to use as an SQL identifier. This allows each data source to be uniquely identified when performing heterogeneous operations.

Data services requiring logins must be associated with a domain in the Authentication Server. When users connect to the data service through a data source name, the domain name is used to retrieve user credentials associated with that data service. The credentials are then passed along to the back end database. User credentials are stored in the Authentication Server. For more information on personal and shared logins, see [Data Source Names \(DSNs\)](#).

Data services can also contain optional information to control Federation Server driver behavior, such as locking semantics and tracing. Data services form the foundation for connectivity to a source of data, and privileges may be assigned to data services to control which users may connect to the given data service.

The Federation Server supports disparate data services by providing software in the form of Federation Server drivers, which access the physical data that an application processes. There are two types of drivers:

- SASProprietary drivers provide access to SASProprietary file formats, such as SAS data sets. These drivers are in-process drivers that access local SAS data in the server process. SASProprietary drivers are installed with Federation Server.

- Remote Federation Server drivers provide access to third-party relational databases, such as DB2®, MySQL®, ODBC databases, Oracle, and Teradata®, by connecting to a remote server process. Remote Federation Server drivers are installed when those products are licensed.



**Note:** Data service names are case-insensitive.

## Supported Data Services

The Federation Server supports the following data services: SAS data set, SAP®, and several third-party relational databases.

### Supported Data Service Types

The following table lists the supported data service types:

Data Sources	Data Service Types	Default Drivers	Supported Drivers
Base	BASE	BASE	BASE
DB2	DB2UNXPC	DB2	ODBC, DB2
MySQL	MYSQL	MYSQL	ODBC, MYSQL
Oracle	ORACLE	ORACLE	ODBC, ORACLE
SAP	SAP	SAP	SAP
SQL Server	SQLSERVER or SQLSVR	ODBC	ODBC
Teradata	TERADATA	TERADATA	ODBC, TERADATA

### SAS Data Set

The SAS data set is the Base SASProprietary file format for SAS software, which contains data values that are organized as a table of observations (rows) and variables (columns). The supported file format is the same as SAS data sets that are created by the BASE engine for Version 7 and later. A supported SAS data set uses the extension .sas7bdat.

The DataFlux Driver for SAS BASE Library provides read and update access to legacy SAS data sets. In addition, the driver creates SAS data sets that can be accessed by both the Federation Server and Base SAS software. The driver supports standard Base SAS storage functionality such as indexing, general integrity constraints, and SAS formats and informat. The driver also supports much of the Federation Server TSSQL (TSSQL) functionality. For more information about supported functionality and compatibility guidelines, see [SAS Data Set Reference](#).

### SAP

For more information about supported functionality and compatibility guidelines, see [SAP Reference](#).

### Third-Party Relational Databases

The Federation Server can access data in several third-party relational databases, including DB2, MySQL, ODBC databases, Oracle, and Teradata.

The relational database drivers read, update, and create tables for those third-party relational databases on behalf of the Federation Server client. Each driver supports most of

the TSSQL functionality. The Federation Server drivers support native database functionality by using the SQL dialect that is implemented by the third-party databases. For details about supported functionality and compatibility guidelines, see the specific data service reference:

- [DB2 Reference](#)
- [MySQL Reference](#)
- [ODBC Reference](#)
- [Oracle Reference](#)
- [Teradata Reference](#)

## Data Source Names (DSNs)

After configuring data services, an administrator will create a data source name (DSN). DSNs are used to expose data services to connecting Federation Server users. Typically, when a client connects to the Federation Server, they will specify a DSN. The Federation Server administrator can assign privileges to determine which users may connect. In order to connect, a user must be granted Connect privilege on either the Federation Server, a specific data service, or a specific DSN.

A DSN must reference a data service to which it will connect. The DSN will define how SQL security will be enforced. It can be configured so that the Federation Server enforces SQL privileges defined for the data service.



**Note:** Data source security is not bypassed; Federation Server security is applied in addition to the data source security. For more information, see [Authorization and Security](#).

If Federation Server SQL security is enabled, the TSSQL driver is also required, and the SQL dialect is automatically set to TSSQL. If Federation Server SQL security is disabled, the administrator can choose either TSSQL dialect or data source (native) dialect. For example, if you are connected to Oracle, then native dialect would be the SQL supported by Oracle. The SQL dialect for Base data services is always TSSQL.

For data services requiring credentials, the administrator may also use the DSN to configure how database logins are retrieved. The DSN can be configured to use the personal credentials of the user, or retrieve the login from a shared login. For more information on shared logins, see the *DataFlux Authentication Server Administrator's Guide*. If you are using a shared login, you can optionally specify a consumer group from the DSN. This is only required to make it clear which shared login to use if multiple shared logins are available in the same domain to connecting users.



**Note:** Data source names are case-insensitive.

## Shared Login Configuration

If the Federation Server is used to access shared logins via DSN setup, then the server must be configured appropriately. This involves defining a shared login key and a shared login manager login. The shared login key identifies which shared logins that are created in the Authentication Server are available to this Federation Server instance. The key defined in the Federation Server must match the key that is part of the shared login definition in the Authentication Server.

The shared login manager login identifies the credentials used to connect to the Authentication Server and retrieve the shared login. The shared login manager login must identify a user that is a manager or owner of that shared login. The shared login is not directly readable by consumers of the shared login; it can only be read by managers or the owner. The shared login manager's password is encrypted.

## Shared Login Conflict Resolution

Shared logins are created in the Authentication Server when there is a need to share logins among multiple users. Logins must be owned by users, not groups. Therefore, shared logins are the only mechanism to provide this functionality.

Shared logins consist mainly of the login (and domain) to share, and the consumers who may use that login. The consumers will typically list one or more groups, which is a best practice. The conflict arises when a particular user is in a consumer group (directly or indirectly) of multiple principal maps for the same domain. The following scenarios explain how shared login conflicts are resolved.

### Scenario 1: Application Users

In the following scenario, an application exists which requires the use of a particular set of database credentials to access its protected data. For example, an HR application has data content stored in Oracle and DB2. The Federation Server administrator will manage credential by completing the following steps:

1. Identifying all the users of the HR application. The users may have different roles or data access privileges, but they all need to access the data. These users (or subgroups) will all be placed in the group HR\_USERS.
2. Creating a shared login for each domain. In this case, the administrator would create an HR\_ORACLE and HR\_DB2 shared login. For both shared logins, the administrator would specify the HR\_USERS group as a consumer member of the shared login. Each shared login would contain the appropriate principal and domain for the database.
3. Specifying the GROUP option to qualify the users with the shared login, either in the DSN itself, or on the connection string specifying the DSN. In this case, the GROUP option would be HR\_USERS.
4. Setting authorizations on different users and groups to control which set of users can perform which operations, for example Select versus Update versus Delete. All of the users and groups should be members of the HR\_USERS group.

At Connect time, the HR\_USERS group is used to identify the correct shared login for each underlying database connection. If the connecting user were a consuming member of another shared login, the GROUP value would properly identify which one to use.

### Algorithm When Using the GROUP Option

Shared logins are initially considered candidates for outbound credentials selection if the domain and shared login key match. If the domain is empty, shared logins for any domain initially qualify. This also applies to the shared login key, which is configured as part of the Federation Server.

If the GROUP connection string option is specified (which is derived from the consumer group in the DSN configuration), then only maps where the group is a direct or indirect

consumer will be considered a candidate for outbound credentials selection. The basic algorithm selects a map based on the closeness of the specified group to the map.

### **Candidate Map Processing**

1. If the user is not a direct or indirect member of the shared login consumer group, the map is not a candidate; otherwise,
2. If the GROUP is not a direct or indirect consumer of the map, the map is not a candidate; otherwise,
3. The distance from the GROUP to the map is computed by following the group member-of relationship up to the group that is the direct consumer. The candidate map is retained if and only if the distance is less than or equal to the current minimum distance to the map. The current minimum distance is updated.
4. After all candidate maps have been processed,
  - a. if exactly one has been retained, return OK and the associated credentials; or
  - b. if two or more were retained, check the closest two, and return the credentials associated with the closer of the two or return an error if the distances are the same unresolved conflict; or
  - c. if no maps have been retained, then return OK but empty credentials.

### **Scenario 2: Organized Consuming Users**

In the following scenario, the administrator has organized the users based on company organization or another classification. The administrator wants to use this relationship to have users qualify for a particular shared login, for example:

- The administrator wants to grant access to Oracle account EXECUTIVE\_USER to his most privileged users, identified by the MARKETING\_EXECUTIVE group.
- The administrator wants to grant access to Oracle account MARKETING\_USER to members of the marketing division in the company, identified by the MARKETING group.
- The administrator wants to grant access to Oracle account STANDARD\_USER to all other known users in the system, identified by the USERS group.
- The administrator has created groups which reflect the company's organizational chart. The MARKETING group reflects all members of the marketing organization, with the MARKETING\_EXECUTIVE group being a member of the MARKETING group.

The administrator creates shared logins for the EXECUTIVE\_USER, MARKETING\_USER and STANDARD\_USER Oracle accounts. Next, he assigns MARKETING\_EXECUTIVE, MARKETING and USERS consuming groups, respectively, to these shared logins. Then, the shared login chosen would be as follows:

- For members of the MARKETING\_EXECUTIVE group, they would be closest to the shared login identified by that group, even though they were likewise members of the MARKETING and USERS groups. Therefore, this set of users would consume the EXECUTIVE\_USER Oracle account.
- For members of the MARKETING group, they would be closest to the Shared Login identified by that group, even though they were likewise members of the USERS

group. Therefore, this set of users would consume the `MARKETING_USER` Oracle account.

- All other known users would only qualify for the shared login identified by the `USERS` group. Therefore, this set of users would consume the `STANDARD_USER` Oracle account.

In this scenario, the administrator would not use the `GROUP` option, since the option accepts only a single value, and no single value works for all users. The administrator would omit the `GROUP` option and allow a closeness algorithm to identify which shared login to use.

### **Algorithm When No `GROUP` Option is Specified**

If the `GROUP` connection string option is not specified, then all maps where the user is a direct or indirect consumer will be considered a candidate for outbound credentials selection. The basic algorithm selects a map based on the closeness of the specified user to the map, or `USERS` if the user is not a consumer, and finally `PUBLIC` if `USERS` isn't either. If the user is `PUBLIC`, selection is done based on `PUBLIC` alone.

### **Candidate Map Processing For a User**

1. If the user is not a direct or indirect consumer of the map, the map is not a candidate; otherwise,
2. The distance from the user to the map is computed by following the group member-of relationship up to the group that is the direct consumer. The candidate map is retained if and only if the distance is less than or equal to the current minimum distance to the map. The current minimum distance is updated.
3. After all candidate maps have been processed,
  - a. if exactly one has been retained, return the associated credentials; or
  - b. if two or more were retained, check the closest two, and return the credentials associated with the closer of the two or return an error if the distances are the same (unresolved conflict); or
  - c. Continue on to [Candidate Map Processing For `USERS`](#). If the user is `PUBLIC`, go to [Candidate Map Processing For `PUBLIC`](#).

### **Candidate Map Processing For `USERS`**

1. If `USERS` is a direct consumer of the map, then the candidate map is retained.
2. If a candidate map has been retained already, return an error (unresolved conflict).
3. After all candidate maps have been processed, if exactly one has been retained, return OK and the associated credentials; or
4. Continue on to [Candidate Map Processing For `PUBLIC`](#).

### **Candidate Map Processing For `PUBLIC`**

1. If `USERS` is a direct consumer of the map, then the candidate map is retained.
2. If a candidate map has been retained already, return an error (unresolved conflict).

3. After all candidate maps have been processed, if exactly one has been retained, return OK and the associated credentials; or
4. Return OK but empty credentials if no candidate maps retained.

### **Path Length Computation Details**

If USERS or PUBLIC is a member of another group, and that group is a map consumer, direct or indirect, the path length is not incremented when traversing from the user to the map. For the purposes of map selection, this effectively makes placing either of these two groups in another group a shorthand way to place all users in that group.

## **Users, Groups, and Roles**

Certain aspects of the Federation Server require knowledge about users and groups, specifically authorization enforcement and login retrieval for database access. Users and groups are defined and maintained through the Authentication Server. The Federation Server Manager will connect to the Authentication Server as well as the Federation Server, so it can present user and group selections needed for authorization definitions.

Database logins for users are also managed through the Authentication Server. The Federation Server will access the appropriate logins based on the domain associated with the data service and DSN used to connect to the data. Logins can be owned by individual users, or shared among many users through shared logins. For more information on creating and configuring shared logins, see the *DataFlux Authentication Server Administrator's Guide*.

### **Predefined Roles**

Federation Server has one predefined role: ADMINISTRATOR. Users who are granted administrator privilege on the Federation Server are automatically assigned the ADMINISTRATOR role. Administrators have the authority to perform the following actions:

- create and drop data source names (DSN)
- grant and deny privileges
- create and drop data services, catalogs, and schemas

Administrators are automatically granted all privileges on all objects.

### **Predefined Users**

Federation Server has one predefined user: SYSTEM. SYSTEM user accounts are specified in the Federation Server configuration file.

SYSTEM users are defined in the Federation Server configuration file, `dfs_serv.xml`. The user account that the Federation Server is running under is automatically considered a SYSTEM user, even if the account is not specified in the `dfs_serv.xml` file.

SYSTEM users are automatically granted all privileges on all objects, and can also perform all actions.



**Best Practice:** Refer to [Best Practices - The SYSTEM user should create and define the Federation Server administrators](#) for additional information.

## Orphaned Accounts

Orphaned accounts are user or group accounts that have been deleted on the Authentication Server, but the reference to the accounts still exist on the Federation Server.

For example, a user can be defined on the Authentication Server and own a DSN on the Federation Server. Then, an administrator deletes the user from the Authentication Server. The Federation Server does not recognize this user deletion. The Federation Server still has a record of the user, and the DSN owner is still the same user, even though the user was deleted. The Authentication Server and Federation Server will be out of synch until those references are cleaned up on the Federation Server.

As the administrator, you need to remove the records for the accounts and manually reassign ownership of objects, such as the DSN, to another user. For more information on how to remove orphaned accounts, see the *DataFlux Federation Server User's Guide*.

## Authorization and Security

Properly configured security for the Federation Server ensures that both the server and its data are secure. Data is protected against unauthorized access, and can be guaranteed secure transmission lines for transferring data. The security features are flexible with respect to the types and amount of security, and both security setup and maintenance are easily managed.

The Federation Server supports the following security features:

- Authorization is the process that determines which users have which authorizations to which resources.
- Encryption is the act or process of converting data to a form that only the intended recipient can read or use.

## Authorization

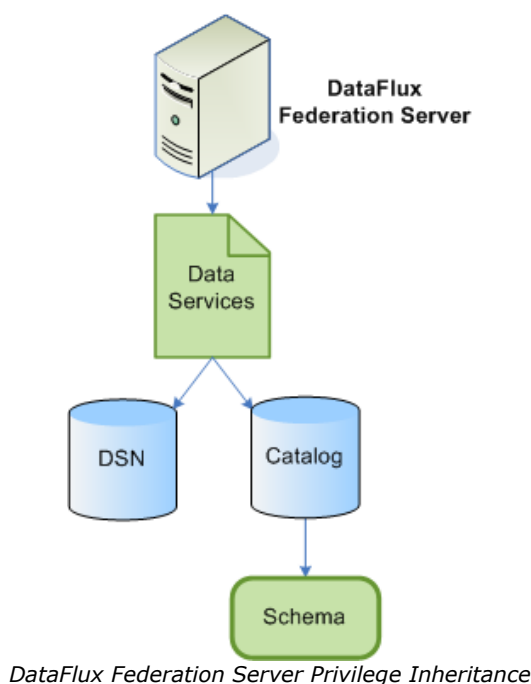
Authorization determines which users have which authorizations to which resources. The Federation Server implements its own authorization provider. Whether enabled or disabled, the security that is available for the data source is also supported.

- **Federation Server authorization** — the authorization process that uses an internal database to store security definitions for users and objects. The outcome of the authorization process is a decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.
- **Data source security** — the authorization process that is provided by the underlying data source. The security that is available for a third-party database such as Oracle is an example of data source security. The Federation Server establishes the connection to the database but makes no attempt to enforce its own security over the security provided by the data source.

A DSN definition configures the authorization that is enforced for data access. The available authorization process depends on the data source. SAS data sets require Federation Server authorization; third-party databases enforce data source security and can be additionally configured for Federation Server authorization.

DataFlux authorization is enforced by the Federation Server by using an internal database that stores security definitions created by the administrator. Authorization definitions consist of privileges that grant or deny authorization to users and groups on particular objects, like data service, catalog, and schema. Many privileges are SQL based, such as Select, Insert, Update, Delete, Create Table, and Drop Table. Other privileges include Administer, Create DSN, Connect (you can connect to the data service), and Trace (you can create trace files).

The Federation Server supports inheritance for privileges. The following shows privilege inheritance between different Federation Server objects:



The diagram illustrates the following inheritance model:

- Schemas inherit privileges from catalogs
- Catalogs and DSNs inherit privileges from data services
- Data services inherit privileges from the Federation Server

A privilege on the DSN overrides a privilege on the data service or Federation Server. All SQL privileges are relevant to the Federation Server, data service, catalog, and schema and follow the inheritance settings above.

The following table describes the privileges supported by the Federation Server:

Supported Privilege	Description
Administer	Controls the ability to configure the server. This privilege can only be set on the Federation Server. Users who are granted Administer privilege are automatically granted all other privileges.

Supported Privilege	Description
Alter Table	Controls the ability to add or drop columns in a table or create or drop indexes with the ALTER TABLE statement. The authorization is set on the server, catalog, schema objects, and data service.
Connect	Controls the ability of the user to connect, using either the DSN or data service. This privilege is supported by the Federation Server, data service, and DSN.
Create Table	Controls the ability to create new tables or views with the CREATE TABLE or CREATE VIEW statement. The authorization is set on the server, catalog, schema objects, and data service.
Delete	Controls the ability to delete data with the DELETE statement or analogous method call. The authorization is set on the server, catalog, schema objects, and data service.
Drop Table	Controls the ability to remove tables or views with the DROP TABLE or DROP VIEW statement. The authorization is set on the server, catalog, schema objects, and data service.
Insert	Controls the ability to add data with the INSERT statement or analogous method call. The authorization is set on the server, catalog, schema objects, and data service.
References	Controls the ability to create a foreign key reference to an existing table. The authorization is set on the server, catalog, schema objects, and data service.
Select	Controls the ability to retrieve data with the SELECT statement. The authorization is set on the server, catalog, and schema objects.
Trace	Controls the ability of the user to enable tracing and create trace files. This privilege can only be set on the Federation Server.
Update	Controls the ability to modify data with the UPDATE statement or analogous method call. The authorization is set on the server, catalog, schema objects, and data service.
Create DSN	Controls the ability of the user to create a DSN. The authorization can be set on the data service or the Federation Server.

By default, users are not granted any privileges. The SYSTEM user or Federation Server administrator must grant privileges in order for users to perform actions and gain access to data. The Federation Server administrator is a user who has been granted Administer privileges on the Federation Server.

## Data Source Security

Relational databases provide authorization that limits the operations that can be performed on their data. The Federation Server respects authorizations that are defined and enforced on a third-party database and allows connections to be configured to use a database's security exclusively.

The type of authorization that is enforced depends on the data source. For example, third-party database servers such as Oracle and DB2 have specific database security.

When an administrator configures a DSN definition to use data source security, it means that the authorization is enforced by the database. The Federation Server establishes the connection to the database but makes no attempt to secure the data that is available through the connection.

## Privilege Determination Summary

To summarize how privileges are determined for a user, the following algorithm is used:

- SYSTEM and Administrator users may not be denied access, and therefore are always granted access.
- For other users, a specific privilege is first looked up at the current object in the hierarchy, in the following order:
  1. On the current object, the privilege is first searched for under the identification of the specific user. If a specific determination is made (Grant or Deny), then that type is returned for the privilege, and privilege lookup stops.
  2. On the current object, the privilege is searched for under the first level of group membership. For more information, see [Group Membership](#). At this level, if any group indicates Deny, then Deny is returned. If all groups indicate Grant, then Grant is returned.
  3. On the current object, the privilege is searched for under the next level of group membership, per the same rules as the previous bullet. If no specific determination is made, repeat for all levels of group membership.
  4. If no determination is made on the current object, then privilege determination goes to the next higher level in the object hierarchy. For more information, see [Inheritance](#). The privilege search algorithm repeats as above, first under the identification of the specific user, and then group membership.
  5. If all objects in the hierarchy are searched and no privilege determination is made, then a Deny is returned for the privilege type for the user.

## Group Membership

The Federation Server will support all the groups in the Authentication Server, plus two static (fixed membership) groups: PUBLIC and USERS. If a user's authentication credentials can be located for a registered user, then that user is a member of the USERS group; otherwise, the user is a member of the PUBLIC group. The USERS group is a member of the PUBLIC group.

Privileges can be granted to and denied from groups as well as individual users. Users who are members of a group will inherit the privileges from the group unless explicitly overridden on the individual user.

### Notes on Groups

Groups can present some conflict problems during privilege resolution. For example, UserA is a member of Group1 and Group2. Group1 is granted a privilege, while Group2 is denied the same privilege. The following conditions determine whether UserA is granted or denied the privilege:

1. The conflict exists for group membership at the same level only. Closer group memberships take precedence over more distant ones.

2. A conflict at the same group precedence level results in a Deny.
3. USERS takes precedence over PUBLIC, since USERS is a member of PUBLIC.

## Inheritance

The following rules apply to inheritance:

- Schema inherits from catalog
- Catalog inherits from data service
- DSN inherits from data service
- Data service inherits from the Federation Server

## Database Functionality Supported by Drivers

Because the Federation Server supports several data services, a broad range of database functionality that is unique to each data service is provided. For example, while a particular data service provides transaction support, a different data service might not provide transactions but supports indexes and integrity constraints.

You must understand database functionality and how its implementation affects processing, performance, and integrity of your data in order to determine which data services are most appropriate for different types of applications. Because database functionality is unique to each data service, you cannot make assumptions about the data service to be accessed. For example, an application cannot request a locking level just because that locking level is more efficient. An application must respond to the attributes of a Federation Server driver.

Database functionality is applied through the Federation Server driver when the application submits requests, which can be in the form of TSSQL statements or the SQL statements that are the implementation of the data service.

The following topics describe database functionality. For information about the database functionality that is supported by each data service, see the database functionality support topic for each data service in the data service reference.

## Bookmarks

Bookmarks are stored locations that are used for quick retrieval. A bookmark identifies a row of data in a result set so that an application can quickly return to the row, which performs speed reads, updates, and delete operations. In addition, bookmarks are used for scrollable cursor functionality in data services that do not support scrollable cursors.

Bookmark functionality is implemented differently for each data service. The stored value might be a row number or a disk address. However, not all data services support bookmarks.

## Bulk Operations

A bulk operation is the ability to rapidly and efficiently add, delete, or update large amounts of data through one operation. For example, a bulk copy operation copies large amounts of data, and a bulk load operation adds large amounts of data into a data service.

Typically, methods to do bulk operations are specific to a particular database. The Federation Server uses the data service's highest performance bulk operations. For example, requesting a bulk add operation for an Oracle table causes the Oracle Bulk Loader to be invoked.

Bulk operations are requested by using TSSQL statements as well as by using the specific Federation Server driver.

## **Cursor Support**

A cursor establishes or keeps track of the position in a result set and provides scrolling capabilities to traverse it. For example, when a table is opened, the cursor points to the first row in the file. Various commands then move the cursor forward, backward, to the top of the file, to the bottom of the file, and so on.

A scrollable cursor enables you to back up and revisit a row, start at the end of the file and work backward, skip some rows, or go directly to a specific row. A non-scrollable cursor, such as a forward-only cursor, moves forward only through the result set.

The decision about what cursor functionality is needed is determined by the level of performance and the application's data access requirements.

## **Indexes**

An index is a component of a database that contains information about the physical location of data that is stored in a file. An index establishes key order on the data so that operations that access data in a particular order perform better. For example, rather than search the entire file, an application can use an index to locate the data quickly and efficiently. Indexing columns that are frequently used by an application, especially for large files, greatly increases the speed of queries.

Because databases implement different types of indexes, data services that are supported by the Federation Server provide different types of indexes.

## **Integrity Constraints**

Integrity constraints ensure the accuracy and consistency of data in a relational database. Integrity constraints prevent incorrect insertion, modification, or destruction of data in a database.

Because databases implement different types of integrity constraints, data services that are supported by the Federation Server provide different types of integrity constraints such as general and referential integrity constraints.

## **Locking**

Locking is the process by which a database protects data from simultaneous updates in a multi-user environment.

Locking levels prevent applications from reading data that is being changed by other applications and prevents multiple users from changing the same data at the same time. Locking is an essential part of every system that allows data to be updated by multiple users or applications.

Locking behavior can be controlled in an application. These are the various locking levels that are supported by the Federation Server data services:

- **file-level** — locking controls concurrent access at the file level. File-level locking enables read access to multiple users but restricts all other access to the file when a query requests update access. When an application opens a file for update, it is locked until the application releases the lock. All other applications must wait until the lock is released to continue processing.
- **row-level locking** — controls concurrent access at the row level. Row-level locking enables concurrent read access and update access to the file by multiple users but prevents concurrent update access to the same row. There are two types of row-level locking: optimistic locking and pessimistic locking.

Optimistic locking assumes that few or no update requests will be made by other applications, and therefore provides less intensive checking for simultaneous updates.

Pessimistic locking assumes that many updates might be requested for a particular row and therefore provides additional functionality that ensures that no updates were made simultaneously.

- **column-level (values) locking** — controls concurrent access at the column level. Column-level locking enables concurrent read access and update access to the file by multiple users but prevents concurrent update access to the same column.
- **read locking** — enables read access to multiple users but restricts all other access. That is, the data cannot be altered. A read lock is the fastest type of lock, because the server does not maintain a lock on the data.

The Federation Server data services provide different levels of locking. For example, some data services such as SAS data sets provide file-level locking, while other data services, such as the third-party databases DB2 and Oracle, provide row-level locking.

Selecting the appropriate locking level can greatly improve an application's performance.

For information about the locking levels that are provided by each data service, see the database functionality support topic for each data service in the data service reference.

## Threaded I/O Processing

I/O (input/output) processing is the ability to read a segment of data from a storage device (such as disk) and transfer the data to memory or, conversely, to transfer the data from memory and write it to the storage device. Threaded I/O processing fetches data in multiple threads to process a query such as a SELECT statement. Fetching data by using multiple threads provides an application an efficient method of retrieving data, which boosts performance for applications that might process the data faster than the data can be fetched. This is a bottleneck condition that is referred to as being I/O bound.

I/O requests to DB2 and ODBC database data services can be processed in threads.

When applications combine the parallel processing services that are provided by the Federation Server languages with a data service that provides threaded I/O, the partnership further enhances performance.

## Transaction Support

Transaction support is functionality by which data is protected by ensuring that updates are either fully applied or rolled back to the pre-transaction state in the event of an interrupted operation.

A transaction is an atomic unit of work. That is, a transaction either completely succeeds or has no effect. After a logical, consistent set of changes has occurred, a transaction is ended either by committing the changes, which makes them permanent, or canceling the changes, which returns the values changed by a transaction to their original state.

A commit is an operation that makes updates permanent. When a connection to a database is established, auto-commit is set to on by default, which means that each individual SQL statement is treated as a transaction. As soon as the statement is executed, if no return code is detected, the transaction is automatically committed. If update problems are detected, the application initiates a rollback of the transaction.



**Note:** To allow a transaction to be made up of multiple SQL statements, the application must turn off auto-commit, execute and commit the collection of statements, and then turn auto-commit back on (which reactivates the default of one SQL statement per transaction).

A rollback is the process that reverses the current transaction out of the database, returning the data to its former state. A rollback is performed when processing a transaction fails at some point, and it is necessary to start over.

Transaction support involves the ability to commit updates or rollback updates and then reapply them, often from a transaction log, to recover from a database crash. The rollback and roll forward (reapplication of transactions) automatically occur when the database is restarted.

Some applications have a requirement to treat groups of updates to a particular data service as a single unit. That is, either the entire group of updates is applied to the data or none of them is applied. Applications can test for errors and execute specific commit and rollback operations as needed. TSSQL supports both COMMIT and ROLLBACK statements.

Not all data services provide transaction management. For example, transaction management is not available for SAS data sets. However, transaction support is available for third-party relational databases like DB2 and Oracle. Many of the third-party relational databases also provide transaction support in their native SQL dialect.



# SAS Data Set Reference

This section provides functionality details and guidelines for SAS® data sets that are supported by the DataFlux® Driver for SAS BASE Library (Driver for BASE).

The SAS data set is a SASProprietary file format, which contains data values that are organized as a table of rows (SAS observations) and columns (SAS variables). The supported file format is the same as a SAS data set that is created by the BASE engine in SAS for Version 7 and later. A supported SAS data set uses the extension .sas7bdat.

The following topics provide additional information about the Driver for BASE:

- [Catalogs and Schemas](#)
- [Data Types](#)
- [Understanding the Driver for BASE](#)
- [Database Functionality Support for SAS Data Sets](#)
- [Data Service Connection Arguments for SAS Data Sets](#)
- [Data Types for SAS Data Sets](#)
- [SAS Column Labels for SAS Data Sets](#)
- [Optimize Performance for SAS Data Sets](#)
- [Indexes for SAS Data Sets](#)
- [Audit Trail for SAS Data Sets](#)
- [Integrity Constraints for SAS Data Sets](#)
- [Generation Data Sets for SAS Data Sets](#)
- [Cross-Environment Data Access for SAS Data Sets](#)
- [Identifier Case Sensitivity for SAS Data Sets](#)
- [Security for SAS Data Sets](#)
- [Numeric Column Length for SAS Data Sets](#)
- [Naming Conventions and Attributes for SAS Data Sets](#)
- [National Language Support](#)
- [TSSQL Statement Table Options](#)

## Catalogs and Schemas

The terms catalog and schema are defined as ANSI SQL standards and refer to the organization of data in a relational database. That is, data is contained in tables, tables are grouped into schemas, and schemas are grouped into catalogs. Catalog and schema names can be used in SQL statements to qualify table references. For example, when querying a database that supports both schemas and catalogs, you can specify a three-level identifier in the form of catalog.schema.table-name.



**Note:** Catalog names are case-insensitive.

## Organize Data with a Catalog

A catalog is a named collection of logically related schemas. The catalog is the first-level (top) grouping mechanism in a data organization hierarchy that qualifies schemas. At least one schema is required for each catalog.

The catalog name can be created and configured using the Federation Server Manager for the BASE data service.

## Organize Data with a Schema

A schema is a data container object that groups logically related objects such as tables and views. The schema provides a unique namespace that is used along with a catalog to qualify names.

For SAS data sets, a schema identifies the physical location such as a UNIX® directory or a Windows® folder that contains a collection of tables. That is, for SAS data, the relationship between a schema and its files is similar to that of an operating system file directory and the files that are contained within that directory. For SAS data, a schema is approximately equivalent to a SAS library.

The schema name can be created and configured using the Federation Server Manager for the Driver for BASE.

## Data Types

A data type, which is an attribute of every column in a table, specifies what type of data the column stores. The data type is the characteristic of a piece of data that says, for example, it is a character string, an integer, a floating-point number, or a date or time. The data type also tells the operating system how much physical storage to set aside for the column.

The following table lists the set of data types that are supported by the DataFlux® Federation Server (Federation Server) for the underlying data services. Note that not all data types are available for storage on each data service. For a list of data types that are available to store data for a particular data service, see [Define Data Types for a Column](#).

Data Type	Description	Storage Size
BIGINT	Stores a large signed, exact whole number, with a precision of 19 digits. The range of integers is -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Integer data types do not store decimal values; fractional portions are discarded.	8 bytes
CHAR( <i>n</i> )	Stores a fixed-length character string, where <i>n</i> is the maximum number of characters to store. The maximum number of characters is required to store each value regardless of the actual size of the value. If char(10) is specified and the character string is only five characters long, the value is right padded with spaces.	<i>n</i> bytes
DATE	Stores a calendar date. A date literal is specified in the format <i>yyyyyy-mm-dd</i> : a four-digit or five-digit year (0001 to 32767), a	8 bytes

Data Type	Description	Storage Size
	<p>two-digit month (01 to 12), and a two-digit day (01 to 31). For example, the date September 24, 1975 is specified as 1975-09-24.</p> <p>The Federation Server complies with ANSI SQL:1999 standards regarding dates. However, not all data services support the full range of dates.</p>	
DOUBLE	Stores a signed, approximate, double-precision, floating-point number. Allows numbers of large magnitude and permits computations that require many digits of precision to the right of the decimal point.	8 bytes
FLOAT( <i>p</i> )	Stores a signed, approximate, single-precision or double-precision, floating-point number. The user-specified precision determines whether the data type stores a single-precision or double-precision number. If the specified precision is equal to or greater than 25, the value is stored as a double-precision number, which is a DOUBLE. If the specified precision is less than 25, the value is stored as a single-precision number, which is a REAL. For example, float(10) specifies to store up to ten digits, which results in a REAL data type.	4 or 8 bytes
INTEGER	Stores a regular size signed, exact whole number, with a precision of ten digits. The range of integers is -2,147,483,648 to 2,147,483,647. Integer data types do not store decimal values; fractional portions are discarded.	4 bytes
NCHAR( <i>n</i> )	Stores a fixed-length character string like CHAR but uses a Unicode national character set, where <i>n</i> is the maximum number of multibyte characters to store. Depending on the platform, Unicode characters use either two or four bytes per character and support all international characters.	<i>n</i> bytes
NVARCHAR( <i>n</i> )	Stores a varying-length character string like VARCHAR but uses a Unicode national character set, where <i>n</i> is the maximum number of multibyte characters to store. Depending on the platform, Unicode characters use either two or four bytes per character and can support all international characters.	<i>n</i> bytes
REAL	Stores a signed, approximate, single-precision, floating-point number.	4 bytes
SMALLINT	Stores a small signed, exact whole number, with a precision of five digits. The range of integers is -32,768 to 32,767. Integer data types do not store decimal values; fractional portions are discarded.	2 bytes
TIME( <i>p</i> )	Stores a time value. A time literal is specified in the format <i>hh:mm:ss[.nnnnnnnnnn]</i> ; a two-digit hour 00 to 23, a two-digit minute 00 to 59, and a two-digit second 00 to 61 (supports leap seconds), with an optional fraction value. For example, the time 6:30 a.m. is specified as 06:30:00. When supported by a data service, the <i>p</i> parameter specifies the seconds precision, which is an optional fraction value that is up to nine digits long.	8 bytes
TIMESTAMP( <i>p</i> )	Stores both date and time values. A timestamp literal is specified	8 bytes

Data Type	Description	Storage Size
	in the format <i>yyyy-mm-dd:hh:mm:ss[.nnnnnnnnnn]</i> : a four-digit year 0001 to 9999, a two-digit month 01 to 12, a two-digit day 01 to 31, a two-digit hour 00 to 23, a two-digit minute 00 to 59, and a two-digit second 00 to 61 (supports leap seconds), with an optional fraction value. For example, the date and time September 24, 1975 6:30 a.m. is specified as 1975-09-24:06:30:00. When supported by a data service, the <i>p</i> parameter specifies the seconds precision, which is an optional fraction value that is up to nine digits long.	
TINYINT	Stores a very small signed, exact whole number, with a precision of three digits. The range of integers is -128 to 127. Integer data types do not store decimal values; fractional portions are discarded.	1 byte
VARBINARY( <i>n</i> )	Stores varying-length binary data, where <i>n</i> is the maximum number of bytes to store. The maximum number of bytes is not required to store each value. If varbinary(10) is specified and the binary string uses only five bytes, only five bytes are stored in the column.	
VARCHAR( <i>n</i> )	Stores a varying-length character string, where <i>n</i> is the maximum number of characters to store. The maximum number of characters is not required to store each value. If varchar(10) is specified and the character string is only five characters long, only five characters are stored in the column.	<i>n</i> bytes

## Data Type Characteristics

The Federation Server supports several data types: numeric, character, and date and time.

### Numeric Data Types

Numeric data types store numbers (for example, quantities and currency values). The choice of a numeric data type depends on the type of number that is being stored and on how the number will be used.

Characteristics to consider when choosing a numeric data type include the following:

- Whether you want to use exact numeric data types or approximate numeric data types. Exact numeric data types such as the integer data types represent a value exactly. Approximate numeric data types, such as REAL, DOUBLE, and FLOAT, do not store the exact values that are specified for many numbers. For many applications, the tiny difference between the specified value and the stored approximation is not noticeable unless exact numeric behavior is required.
- Whether you want to store whole numbers or decimal numbers. Note that for integer data types, if a decimal value is inserted, the fractional portion is discarded. For example, if 2.7 is inserted, the stored value is 2.
- How to store numbers with decimal points. Floating-point format is a number in which the decimal point is not fixed. A floating-point data type is used for longer decimal values and for quickly calculating a large range of numbers.

For the floating-point data types, if you enter an integer that is too large, an error occurs. Enter the value using floating-point notation.

- The precision for a data type, which specifies the total number of digits that a number can contain. The more digits, the higher the precision for the value. If a value is out of range, an error occurs. For example, for the BIGINT data type, an error occurs for a value that is larger than 9,223,372,036,854,775,807.

To avoid errors or incorrect results, you must consider the results when performing operations on numeric values, particularly for the integer data types. For example, for the INTEGER data type, which has a precision of 10 digits, if you multiply two large integers (say,  $33432 \times 79879$ ) and the result is larger than 2,147,483,647, an overflow error occurs. If you expect a result that is larger than the data type's precision, you can assign the result as a larger data type such as DOUBLE or BIGINT, or you can enter the expression as a double constant (say,  $33432.0 \times 79879.0$ ), not an integer, in order to force the expression to be evaluated as a double precision, floating-point number.

## Character Data Types

The Federation Server provides several character data types that store character string (text) data. Character data types can contain alphabetic characters, numeric digits 0 through 9, and other special characters. Although a character string can include numbers, these numbers cannot be used in calculations.

Each character data type provides a parameter for specifying the maximum number of characters.

The CHAR and VARCHAR data types, by default, use the character set that is determined by the Federation Server. The NCHAR and NVARCHAR data types use a Unicode character set.

## Date and Time Data Types

The Federation Server supports several data types for the specific purpose of storing dates and times. The date and time ranges are data service specific.

## Defining Data Types for a Column

When defining a data type, you must use the SQL data type keywords for either the data types supported by the Federation Server languages or the data types that are supported by the target database SQL language.

If you submit TSSQL statements, use the Federation Server data type keywords. For information about how to define data types for a column by using TSSQL, see the *DataFlux Federation Server: TSSQL Language Reference*. If, however, you submit the native SQL dialect that is implemented by a third-party database, use the data type names for the target database. For information about how to define data types by using the SQL language for a specific data service, see the database documentation.

When defining data types by using the Federation Server languages, keep in mind that for data to be stored, the data type must be available for data storage in that data service. Although the Federation Server provides support for several data types, the data types that can be defined for a particular table depend on the data service, because each data service does not necessarily support all of the Federation Server data types. In addition, data services support variations of the standard SQL data types. That is, a specific data type that you specify might default to a different data type and might also have different attributes in

the underlying data service. This occurs when a data service does not natively support a specific data type, but data values of a similar data type can be converted without data loss. For example, to support the INTEGER data type, a SAS data set defaults the data type definition to SAS numeric, which is a DOUBLE. For the Driver for ODBC, if an application requests an ODBC data type such as SMALLINT, but the underlying database does not support that data type, the requested type is mapped to a supported type such as NUMBER for Oracle (which does not support SMALLINT).

The following table lists the Federation Server data types that are available for storing data in a data service. For details about data service implementation for each data type such as data service-dependent attributes, see the appropriate data service reference for data types.

Federation Server Data Type	BASE	DB2®: UNIX and PC	MySQL®	Oracle®	ODBC	Teradata®
BIGINT	✓	✓	✓	✓	✓	✓
CHAR( <i>n</i> )	✓	✓	✓	✓	✓	✓
DATE	✓	✓	✓	✓	✓	✓
DOUBLE	✓	✓	✓	✓	✓	✓
FLOAT( <i>p</i> )	✓	✓	✓	✓	✗	✗
IDENTITY	✗	✗	✗	✗	✗	✗
INTEGER	✓	✓	✓	✓	✓	✓
NCHAR( <i>n</i> )	✗	✗	✗	✓	✓	✗
NVARCHAR( <i>n</i> )	✗	✗	✗	✓	✓	✗
REAL	✓	✓	✓	✓	✗	✗
SMALLINT	✓	✓	✓	✓	✓	✓
TIME( <i>p</i> )	✓	✓	✓	✓	✓	✓
TIMESTAMP( <i>p</i> )	✓	✓	✓	✓	✓	✓
TINYINT	✓	✗	✓	✓	✓	✓
VARBINARY( <i>n</i> )	✗	✓	✗	✓	✗	✓
VARCHAR( <i>n</i> )	✓	✓	✓	✓	✓	✓

✓ Recognized as defined or defaults to a corresponding data type with data service-dependent attributes.

✗ Not available for data storage in that data service. If you attempt to define a column of this data type for the data service, an error occurs.

## Understanding the Driver for BASE

The Driver for BASE is a SASProprietary driver that provides read and update access to legacy SAS data sets and creates SAS data sets that can be accessed by both the legacy and the Federation Server data access services.

The driver supports much of the Base SAS functionality, such as SAS indexing and general integrity constraints, as well as much of the Federation Server TSSQL (TSSQL) functionality.

The Driver for BASE is an in-process driver, which means that it accesses data in the same process that executes the data access services. The driver does not communicate with another process or server to access data.

## Database Functionality Support for SAS Data Sets

The following table lists the database functionality support level for a SAS data set. For general information about database functionality, see [Database Functionality Supported by Data Services](#).

Functionality	Description	Support	Comment
bookmark	Stored location used for quick retrieval and scrollable cursors.	Supports with a SAS data set RID, which is a unique record identifier that identifies each row.	Bookmarks are automatically used in update, delete, and fetch operations in the TSSQL statements.
bulk operations	Ability to add, delete, or update a large quantity of data with one operation.	Automatic by using the TSSQL INSERT statement.	
cursor support	Functionality that keeps track of the position in a result set and enables multiple operations on the result set.	Dynamic and keyset cursors.	Cursor functionality is supported by the driver or by specifying that the Federation Server provides additional cursor support.
index	Component of a database that contains information about the physical location of data stored in a file.	Supports simple and composite indexes. Uses only one index per query.	See <a href="#">Indexes for SAS Data Sets</a> .
integrity constraint	Functionality that ensures correctness, completeness, and compliance of data.	Supports general integrity constraints.	For more information, see <a href="#">Integrity Constraints for SAS Data Sets</a> .
locking	Process by which a database restricts access in a multi-user environment.	File-level and read locking.	Record-level (row-level) locking, which is supported by the BASE engine, is not supported by the Driver for BASE.
multi-row fetch	Enables an application to request more than one row at a time.	Supported.	
stored routines	TSPL programs that can be executed from TSPL or TSSQL programs.	Supported by TSSQL and TSPL language drivers.	
threaded I/O	Efficient method of	Not supported.	

Functionality	Description	Support	Comment
processing	processing a query with multiple threads.		
transactions	Process of protecting data by ensuring the integrity of data during multi-user update sessions.	Not supported.	

## Data Service Connection Arguments for SAS Data Sets

To access data that is hosted on the Federation Server, a client must submit a connection string, which defines how to connect to the data. The data service connection arguments for a SAS data set include connection options and advanced options.

### Connection Options

Connection options are used to establish a connection to a data service. For each connection to the specified data service, one or more connection options should be specified in the create data service data definition language (DDL) statement. For more information on this DDL statement, see [Create Data Service](#).

The following connection options are supported for SAS data sets:

- **CATALOG=catalog-identifier;** — specifies an arbitrary identifier for an SQL catalog, which groups logically related schemas. Any identifier is valid (for example, catalog=base). A catalog name can be up to 32 characters long.

**Requirement:** You must specify a catalog.

- **DRIVER=BASE;** — identifies the data service that you want to access, which is a SAS data set.

**Requirement:** You must specify DRIVER=BASE to access a SAS data set.

- **NAME=schema-identifier;** — specifies an arbitrary identifier for an SQL schema. Any identifier is valid (for example, name=myfiles). The schema identifier is an alias for the physical location of the SAS library, which is much like the Base SAS libref. A schema name must be a valid SAS name and can be up to 32 characters long.

**Requirement:** You must specify a schema identifier.

- **PRIMARYPATH=physical-location;** — specifies the physical location for the SAS library, which is a collection of one or more SAS files. For example, in directory-based operating environments, a SAS library is a group of SAS files that are stored in the same directory.

**Requirement:** You must specify a primary path.

- **SCHEMA=(attributes);** — specifies schema attributes that are specific to a SAS data set. A schema is a data container object that groups tables. The schema contains a



name, which is unique within the catalog that qualifies table names. For a SAS data set, a schema is similar to a SAS library, which is a collection of tables and which has assigned attributes.

## Advanced Options

Advanced driver options are additional options that are not required in order to connect to the data source. They are used to establish connections to catalogs, data source names (DSNs), and schemas. While advanced options can also be used when connecting to a data service, doing so will cause the specified options to apply to all data service connections.

The following advanced options are supported for SAS data sets:

- **ACCESS=READONLY | TEMP;**
  - **READONLY** — assigns a read-only attribute to the schema. You cannot open a SAS data set to update or write new information.
  - **TEMP** — specifies that the SAS data sets be treated as scratch files. That is, the system will not consume CPU cycles to ensure that the files do not become corrupted. **Tip:** Use ACCESS=TEMP to save resources only when the data is recoverable. If TEMP is specified, data in memory might not be written to disk on a regular basis. This saves I/O, but could cause a loss of data if there is a crash.
- **COMPRESS=NO | YES | CHAR | BINARY;** — controls the compression of rows in created SAS data sets.
  - **NO** — specifies that the rows in a newly created SAS data set are uncompressed (fixed-length records). This is the default.
  - **YES | CHAR** — specifies that the rows in a newly created SAS data set are compressed (variable-length records) by using RLE (Run Length Encoding). RLE compresses rows by reducing repeated consecutive characters (including blanks) to two- or three-byte representations. **Tip:** Use this compression algorithm for character data.
  - **BINARY** — specifies that the rows in a newly created SAS data set are compressed (variable-length records) by using RDC (Ross Data Compression). RDC combines run-length encoding and sliding-window compression to compress the file. **Tip:** This method is highly effective for compressing medium to large (several hundred bytes or larger) blocks of binary data (numeric columns). Because the compression function operates on a single record at a time, the record length must be several hundred bytes or larger for effective compression.

**Default:** NO

- **ENCODING=encoding-value;** — overrides and transcodes the encoding for input or output processing of SAS data sets.

**Default:** The default value is the current operating system setting.

- **LOCKTABLE=SHARED | EXCLUSIVE** — places exclusive or shared locks on SAS data sets. You can lock tables only if you are the owner or have been granted the necessary privilege.



- **SHARED** — locks tables in shared mode, allowing other users or processes to read data from the tables, but preventing other users from updating.
- **EXCLUSIVE** — locks tables exclusively, preventing other users from accessing any table that you open.


**Default:** SHARED

## Data Types for SAS Data Sets

When an application submits the TSSQL statements, the Driver for BASE supports the data types that are listed in the following table. For some data type definitions, the Driver for BASE defaults to CHAR, which is a SAS character data type, or DOUBLE, which is a SAS numeric data type. For data types that are not listed in the following table, an error occurs if the data type is defined for a SAS data set. For information about the Federation Server data types, see [Data Service Data Types](#).

The following table lists the data type support for a SAS data set:

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
BIGINT <sup>1 4</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number.   <b>Note:</b> Because DOUBLE is an approximate numeric data type rather than an exact numeric data type like BIGINT, there is potential for loss of precision	DOUBLE
CHAR( <i>n</i> )	CHAR( <i>n</i> )	Specifies a fixed-length character string from 1 to 32,767 bytes; the default is 1. For example, char(100) stores 100 single-byte characters.   <b>Note:</b> CHAR columns for a SAS data set cannot contain ANSI SQL null values.	CHAR( <i>n</i> )
DATE <sup>1 3</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number. By default, applies the DATE9 SAS format, which results in writing or displaying a column's values in the format <i>ddmmmyyyy</i> . Valid SAS date values are in the range from 1582-01-01 to 9999-12-31. Dates outside the SAS date range are not supported and are treated as invalid dates.	DOUBLE

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
DOUBLE <sup>4</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number.	DOUBLE
FLOAT <sup>1 4</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number.	DOUBLE
INTEGER <sup>1 4</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number.	DOUBLE
REAL <sup>1 4</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number.	DOUBLE
SMALLINT <sup>1 4</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number.	DOUBLE
TIME <sup>1 3</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number. By default, applies the TIME8 SAS format, which results in writing or displaying a column's values in the format <i>hh:mm:ss</i> .	DOUBLE
TIMESTAMP <sup>1 3</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number. By default, applies the DATETIME19.2 SAS format, which results in writing or displaying a column's values in the format <i>ddmmyyyy:hh:mm:ss</i> .	DOUBLE
TINYINT <sup>1 4</sup>	DOUBLE	Specifies a 64-bit double precision, floating-point number.	DOUBLE
VARCHAR( <i>n</i> ) <sup>2</sup>	CHAR( <i>n</i> )	<p>Specifies a fixed-length character string from 1 to 32,767 bytes; the default is 1. For example, char(100) stores 100 single-byte characters.</p> <p> <b>Note:</b> CHAR columns for a SAS data set cannot contain ANSI SQL null values.</p>	CHAR( <i>n</i> )

1 Accepts data type keyword but defaults to DOUBLE, which is a SAS numeric data type.

2 Accepts data type keyword but defaults to CHAR, which is a SAS character data type.

3 Because the values are stored as a double precision, floating-point number, you can use the values in arithmetic expressions.

4 Do not apply date and time SAS formats to a numeric data type. For date and time values, use the DATE, TIME, or TIMESTAMP data types.

# SAS Column Labels for SAS Data Sets

A SAS column label is descriptive text that describes the content of a column. The label can be printed in output instead of the column name. A SAS column label can be up to 256 characters long.

The Federation Server support of SAS column labels is determined by the Federation Server driver's implementation. The Driver for BASE supports SAS column labels as follows:

- SAS column labels are supported for all SAS data set data types.
- You can store labels on columns by defining them with the TSSQL CREATE TABLE statement. For example, the following statement stores a label on column EMPSTART:

```
create table base.table1
(empnum double,
 empname char(20),
 empstart double having label 'Employee Start Date');
```

- You can specify a column alias, which is a temporary, alternate name for a column, when you use the TSSQL SELECT statement. Aliases are specified in the SELECT clause to name or rename columns in the result table in order to be clearer or easier to read. For example, the following statement specifies an alias for the column EMPNUM:

```
select empnum as "Employee Number" from base.table1;
```

## Optimize Performance for SAS Data Sets

A SAS data set has several options that you can use to further improve performance. See the information about optimizing system performance in [SAS 9.2 Language Reference: Concepts](#).

## Indexes for SAS Data Sets

A SAS data set supports a simple index, which consists of the values of one column, and a composite index, which consists of the values of more than one column that have been concatenated to form a single value.

In addition to deciding whether you want a simple index or a composite index, you can also limit an index (and its data) to unique values and exclude missing values from the index.

The Driver for BASE supports indexes as follows:

- By using the TSSQL language, you can create an index for any column.
- You can submit the TSSQL CREATE INDEX statement to create an index on columns.
- You can submit the TSSQL DROP INDEX statement to remove a specified index.

For the SAS data set, you must include the FROM table-name syntax to specify the name of the table where the index resides. For a SAS data set, the index file is a SAS file that has the same name as its associated data set and that has a member type of INDEX. There is only one index file per data set; that is, all indexes for a data set are stored in a single file. The index file is considered to be part of the table to which it refers and cannot be specified without reference to its associated table.

- In indexes exist, the TSSQL languages use the indexes for processing the tables. You can also specify `IDXWHERE=` and `IDXNAME=` options in TSSQL.
- Centiles are partially supported. When a SAS data set index is created, statistics called cumulative percentiles (or centiles for short) are automatically stored with the index and refreshed after five percent of the values for an indexed column are updated. The Driver for BASE supports storing centiles, and uses them to determine which index to use in order to optimize a WHERE clause. However, the Driver for BASE and the Federation Server languages do not provide the ability to manipulate centiles (for example, to request that centiles be refreshed or change how often centiles are automatically refreshed).

## Audit Trail for SAS Data Sets

In Base SAS, a SAS data set supports an audit trail, which is an optional SAS file that can be created in order to log modifications to a SAS data set. Each time data is added, deleted, or updated, information is written to the audit trail about who made the modification, what was modified, and when.

The Driver for BASE does not support an audit trail, and the Federation Server languages do not provide functionality for initiating or controlling an audit trail. If a SAS data set has an initiated or suspended audit trail, the Federation Server cannot process the SAS data set for updates. When accessing a SAS data set with an audit trail, opening for input processing provides read-only access and opening for update processing fails.

To update a SAS data set with the Federation Server, you must terminate the event logging and delete the audit file. In Base SAS, use the BASE engine and submit the `AUDIT` statement in the `DATASETS` procedure.

## Integrity Constraints for SAS Data Sets

For a SAS data set, the Driver for BASE supports general integrity constraints, which enable you to restrict the values of columns within a single file. The Driver for BASE supports the check, unique, and primary key types of general constraints. ANSI SQL NOT NULL integrity constraints are supported through the TSSQL `CREATE TABLE` statement.

The Driver for BASE does not support referential integrity constraints. A referential integrity constraint occurs when a primary key integrity constraint in one table is referenced by a foreign key integrity constraint in another table.

## Generation Data Sets for SAS Data Sets

In Base SAS, a SAS data set supports generation data sets, which are archived versions of a SAS data set that is stored as part of a generation group. A generation data set is created each time the file is updated. Each generation data set in a generation group has the same root member name, but each has a different version number. The most recent version of the generation data set is called the base version.

The Driver for BASE does not support generation data sets, and TSSQL does not provide functionality for requesting or specifying generations.

# Cross-Environment Data Access for SAS Data Sets

Cross-environment data access (CEDA) is a Base SAS feature that enables a SAS data set that was created in a directory-based operating environment (for example, UNIX or Windows) to be processed by a different directory-based environment.

CEDA for a SAS data set is supported by the Driver for BASE.


## Identifier Case Sensitivity for SAS Data Sets

The Federation Server languages support both case-sensitive and case-insensitive identifiers. The languages support the following identifier behavior: delimited identifier and non-delimited identifier.

Delimited identifier is a name that is enclosed in double quotation marks. A delimited identifier is case sensitive. The name is preserved as specified and enables the Federation Server to distinguish between identifiers that differ with respect to case.

Non-delimited identifier is an unquoted name. A non-delimited identifier is case insensitive. The name is changed to uppercase.

Identifier behavior for a SAS data set ensures that the file can be processed by the data access services that are provided by both the Federation Server and Base SAS. The following table describes identifier behavior for a SAS data set:

Identifier	Data Service Behavior	Understanding Base SAS Behavior
table name	<p>The Driver for BASE ensures that a table name, whether delimited or non-delimited, can be processed by the data access services that are provided by both the Federation Server and Base SAS. To ensure compatibility, specify all table names in lowercase and non-delimited. Uppercase and mixed case table names are not supported. However, when using the Federation Server languages, you see non-delimited table names return in uppercase, which is appropriate behavior.</p> <p>Duplicate table names depend on whether the operating system is case sensitive or case insensitive. You cannot have duplicate table names on Windows, but they are</p>	<p>Table names are handled in a host-dependent way. On UNIX and Windows, table names are stored on disk in lowercase. When you are retrieving an existing file on UNIX, because UNIX is case sensitive, a table name must be specified in lowercase; on Windows, a table name is normalized to uppercase or mixed case.</p> <p> <b>Note:</b> On UNIX, to avoid unpredictable results between the Federation Server and Base SAS data access services, it is recommended that you do not specify delimited, lowercase table names.</p>

Identifier	Data Service Behavior	Understanding Base SAS Behavior
	acceptable on UNIX.	
column name	You cannot define column names that differ only in case for the same table. This restriction applies to both a delimited identifier and a non-delimited identifier.	Column names are handled the same on all hosts. Column names are stored on disk as specified, such as in mixed case, but column names are not case sensitive. Mixed case is stored and used for presentation purposes only. Therefore, you cannot use the same letters with different combinations of lowercase and uppercase to represent different columns in the same table. You can have MyColumn or myColumn, but you cannot specify both column names in one table. When Base SAS processes column names, however, it internally uppercases them.

For example, on the Federation Server on UNIX the following TSSQL statements specify non-delimited identifiers:

```
create table base.MyTable (MyColumn double);
insert into base.MyTable values (5.0);
insert into base.MyTable values (3.0);
select MyColumn from base.MyTable;
```

Because the identifiers are non-delimited, the Federation Server uppercases specified identifiers, which are stored in the system catalog. However, on UNIX, table names are always stored on disk in lowercase. In the Federation Server, the SELECT statement produces the following output with the column name in uppercase:

MYCOLUMN
5
3

In a UNIX Base SAS session, the following code accesses the SAS data set that was created with TSSQL. Because the table was created without a delimited identifier, either upper or lowercase can be used:

```
libname base '/u/myfiles';

proc print data=base.mytable;
run;
```

The PRINT procedure produces the following output with the column name in uppercase:

Obs	MYCOLUMN
1	5
2	3

## Security for SAS Data Sets

### SAS Passwords

The Driver for BASE supports SAS passwords, which enable you to restrict access to SAS data sets by assigning passwords to the files. You can specify three levels of protection:

- **read** — protects against reading the file.
- **write** — protects against changing the data in the file. Write protection prevents adding, modifying, or deleting rows.
- **alter** — protects against deleting or replacing the entire file. Alter protection also prevents modifying column attributes and creating or deleting indexes.

The Driver for BASE supports assigning passwords when a table is created, and also supports specifying passwords for a password-protected file.

Password-protected files that are created in Base SAS are supported on the Federation Server; password-protected files that are created on the Federation Server are compatible in Base SAS.

To assign or specify a password, submit the ALTER=, PW=, READ=, and WRITE= table options on TSSQL statements. You can specify password table options in TSSQL statements after a specified table name. For example, the following CREATE TABLE statement specifies the PW=table option in order to assign a password to the new SAS data set:

```
create table scores {option pw=luke}
(score1 double, score2 double);
```

The following SELECT statement specifies the PW= option to access the password-protected file:

```
select * from scores {option pw=luke};
```

When specifying SAS passwords, you cannot create a SAS data set that has a SAS password if the physical location for the new SAS data set is secured with either security coupling or a DSN definition that enforces SAS authorization.

## Numeric Column Length for SAS Data Sets

A column's length refers to the number of bytes that are used to store each of the column's values in a file. For example, a column that is defined as a DOUBLE has a storage size of 8 bytes.



In Base SAS, you can control the length with the LENGTH statement in the DATA step. However, the TSSQL language does not support specifying a shorter numeric storage size.

You can read and update a legacy SAS data set that has a shorter numeric storage size, but you cannot specify a shorter numeric size or create a new SAS data set that has a shorter numeric storage size.

## Naming Conventions and Attributes for SAS Data Sets

### Rules for Table and Column Names

Here are rules for SAS data set column and table names:

- SAS table and column names can be up to 32 bytes long. If you specify a name that is longer than 32 bytes, an error occurs.
- The first character in the name must be a letter (A, B, C, . . . , Z) or underscore (\_).
- Subsequent characters can be letters, numeric digits (0, 1, . . . , 9), or underscores.
- You can use uppercase or lowercase letters.
- When creating columns, do not use these names: \_N\_ , \_ERROR\_, \_CHARACTER\_, \_NUMERIC\_, and \_ALL\_.
- When creating tables, do not use these names: \_NULL\_, \_DATA\_, \_LAST\_.
- A column name can contain mixed case. However, to ensure compatibility across SAS software, follow the guidelines in [Identifier Case Sensitivity for SAS Data Sets](#).
- A table name or a column name cannot contain blanks or any special characters other than the underscore unless VALIDVARNAME=ANY.

### Maximum Lengths

Element	Maximum Length
catalog name	32 characters
column label	256 bytes
column name	32 bytes
index name	32 bytes
password	8 bytes
schema name	32 characters
table name	32 bytes

# National Language Support

## Transcoding

The Driver for BASE cannot access a SAS data set that is created with the TRANSCODE=NO attribute, which specifies that character variables cannot be transcoded.

## TSSQL Statement Table Options

Federation Server TSSQL (TSSQL) statement table options specify actions that apply only to the table with which they appear. They enable you to perform operations such as specifying a password for the driver file and renaming columns.

You can specify table options on a TSSQL statement on which you specify a table name, such as the CREATE TABLE, ALTER TABLE, or SELECT statement.

For more information on how to use these options, see "Using TSSQL Statement Table Options" in the *DataFlux Federation Server TSSQL Reference Guide*.

## Syntax

Specify a TSSQL statement table option immediately after the table name, within curly braces { } and including the keyword OPTION. To specify several table options, separate them with spaces or commas.

```
{OPTION option-1=value [... option-n=value] }
```



**Important:** While specifying the syntax for table options, you cannot have a space between the left curly brace { and the keyword OPTION. A space will result in a syntax error.

These examples show table options in TSSQL statements:

```
create table salary {option encrypt=yes read=green};  
select * from salary {option read=green};
```

The following statement table options are supported by the Driver for BASE:

- [ALTER= Table Option](#)
- [BUFNO= Table Option](#)
- [BUFSIZE= Table Option](#)
- [COMPRESS= Table Option](#)
- [ENCODING= Table Option](#)
- [ENCRYPT= Table Option](#)
- [IDXNAME= Table Option](#)
- [IDXWHERE= Table Option](#)
- [LABEL= Table Option](#)

- [LOCKTABLE= Table Option](#)
- [PW= Table Option](#)
- [READ= Table Option](#)
- [REUSE= Table Option](#)
- [TYPE= Table Option](#)
- [WRITE= Table Option](#)

## ALTER= Table Option

Assigns an ALTER password to a SAS data set that prevents users from replacing or deleting the file, and enables access to a read- or write-protected file.

### Syntax

```
ALTER=alter-password
```

### Syntax Description

- ***alter-password*** — Specifies a password. *alter-password* must be a valid SAS name.

### Details

The ALTER= table option applies only to a SAS data set. You can use this option to assign a password or to access a read-protected, write-protected, or alter-protected file. When you replace a data set that is protected with an ALTER password, the new data set inherits the ALTER password.

The password is blotted out when the code is written in the SAS log. For example:

```
from a(alter=XXXXXXX);
```



**Note:** A SAS password does not control access to a SAS file beyond the Driver for BASE or the SAS system. You should use the operating system-supplied utilities and file-system security controls to control access to SAS files outside of the Driver for BASE.

## BUFNO= Table Option

Specifies the number of buffers to be allocated for processing a SAS data set.

### Syntax

```
BUFNO= n | nK | hexX | MIN | MAX
```

### Syntax Description

- ***n* | *nK*** — Specifies the number of buffers in multiples of 1 (bytes); 1,024 (kilobytes). For example, a value of 8 specifies 8 buffers, and a value of 1K specifies 1024 buffers. K must be uppercased.
- ***hexX*** — Specifies the number of buffers as a hexadecimal value. You must specify the value beginning with a number (0-9), followed by an X. For example, the value 2dX sets the number of buffers to 45 buffers. X must be uppercased.

- **MIN** — Sets the minimum number of buffers to 0, which causes the Driver for BASE to use the minimum optimal value for the operating environment. This is the default.
- **MAX** — Sets the number of buffers to the maximum possible number in your operating environment, up to the largest four-byte, signed integer, which is  $2^{31}-1$ , or approximately 2 billion.

## Details

The buffer number is not a permanent attribute of the data set; it is valid only for the current operation.

The BUFNO= table option applies to SAS data sets that are opened for input, output, or update.

A larger number of buffers can speed up execution time by limiting the number of input and output (I/O) operations that are required for a particular SAS data set. However, the improvement in execution time comes at the expense of increased memory consumption.

To reduce I/O operations on a small data set as well as speed execution time, allocate one buffer for each page of data to be processed. This technique is most effective if you read the same observations several times during processing.

## BUFSIZE= Table Option

Specifies the size of a permanent buffer page for an output SAS data set. Use with output data sets only.

## Syntax

```
BUFSIZE= n | nK | nM | nG | hexX | MIN | MAX
```

## Syntax Description

- **n | nK | nM | nG** — Specifies the page size in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); or 1,073,741,824 (gigabytes). For example, a value of 8 specifies a page size of 8 bytes, and a value of 4k specifies a page size of 4096 bytes. K, M, and G must be uppercased.
- **hexX** — Specifies the page size as a hexadecimal value. You must specify the value beginning with a number (0-9), followed by an X. For example, the value 2dx sets the page size to 45 bytes. X must be uppercased.
- **MIN** — Sets the minimum number of buffers to 0, which causes the Driver for BASE to use the minimum optimal value for the operating environment.
- **MAX** — Sets the page size to the maximum possible number in your operating environment, up to the largest four-byte, signed integer, which is  $2^{31}-1$ , or approximately 2 billion bytes.

## Details

The page size is the amount of data that can be transferred for a single I/O operation to one buffer. The page size is a permanent attribute of the data set and is used when the data set is processed.

A larger page size can speed up execution time by reducing the number of times the Driver for BASE has to read from or write to the storage medium. However, the improvement in execution time comes at the cost of increased memory consumption.

To change the page size, copy the data set and either specify a new page or use the Driver for BASE default. To reset the page size to the default value in your operating environment, specify BUFSIZE=0.

Operating Environment Information: The default value for the BUFSIZE= table option is determined by your operating environment and is set to optimize sequential access. To improve performance for direct (random) access, you should change the value for BUFSIZE=.

## COMPRESS= Table Option

Specifies how rows are compressed in a new output data set. Use with output tables only. The category is Table Control.

### Syntax

```
COMPRESS=NO | YES | CHAR | BINARY
```

### Syntax Description

- **NO** — Specifies that the rows in a newly created table are uncompressed (fixed-length records).
- **YES | CHAR** — Specifies that the rows in a newly created table are compressed (variable-length records) by the Driver for BASE using Run Length Encoding (RLE). RLE compresses rows by reducing repeated consecutive characters (including blanks) to two-byte or three-byte representations. Alias is ON. Use this compression algorithm for character data.
- **BINARY** — Specifies that the rows in a newly created table are compressed (variable-length records) by the Driver for BASE using Ross Data Compression (RDC). RDC combines run-length encoding and sliding-window compression to compress the table.

This method is highly effective for compressing medium to large (several hundred bytes or larger) blocks of binary data (numeric variables). Because the compression function operates on a single record at a time, the record length needs to be several hundred bytes or larger for effective compression.

### Details

Compressing a table is a process that reduces the number of bytes required to represent each row. Advantages of compressing a table include reduced storage requirements for the table and fewer I/O operations necessary to read or write to the data during processing. However, more CPU resources are required to read a compressed table (because of the overhead of uncompressing each row), and there are situations where the resulting file size might increase rather than decrease.

Use the COMPRESS= table option to compress an individual table. Specify the option for an output table only, that is, a table name on the CREATE TABLE statement.

After a table is compressed, the setting is a permanent attribute of the table, which means that to change the setting, you must re-create the table. That is, to uncompress a table,

you must drop the table by using the DROP TABLE statement and re-create the table with the CREATE TABLE statement and the COMPRESS=NO option.

## Comparisons

The COMPRESS= table option overrides the COMPRESS= connection string option.

When you create a compressed table, you can also specify the REUSE=YES table option in order to track and reuse space. With REUSE=YES, new rows are inserted in space freed when other rows are updated or deleted. When the default REUSE=NO is in effect, new rows are appended to the existing table.


## ENCRYPT= Table Option

Specifies whether to encrypt an output SAS data set. Use with output tables only.

### Syntax

```
ENCRYPT=YES | NO
```

### Syntax Description

- **YES** — Encrypts the data set. The encryption method uses passwords. At a minimum, you must specify the READ= or the PW= table option at the same time that you specify ENCRYPT=YES. Because the encryption method uses passwords, you cannot change *any* password on an encrypted file without re-creating the table.
  - **NO** — Does not encrypt the data set.
-  **Caution:** Record all passwords. If you forget the password, you cannot reset it without assistance from the Driver for BASE. The process is time-consuming and resource-intensive.

### Details

- Encryption requires approximately the same amount of CPU resources as compression.
- When a SAS data set is encrypted, all associated indexes are also encrypted.

### Examples

This example creates an encrypted data set:

```
create table myfiles.salary {option encrypt=yes read=green}
  name char(15),
  yrsal double,
  bonuspct double);
insert into myfiles.salary values ('Muriel', 34567, 3.2);
insert into myfiles.salary values ('Bjorn', 74644, 2.5);
insert into myfiles.salary values ('Freda', 38755, 4.1);
insert into myfiles.salary values ('Benny', 29855, 3.5);
insert into myfiles.salary values ('Agnetha', 70998, 4.1);
```

To retrieve data from the data set, you must specify the read password:

```
select * from myfiles.salary {option read=green};
```

## ENCODING= Table Option

Specifies the encoding to use for SAS data sets that the BASD code reads and writes. The value for ENCODING= indicates that the SAS data sets have a different encoding from the current operating system setting.

### Syntax

```
ENCODING='encoding-value'
```

### Syntax Description

- **encoding-value** — Specifies the encoding value to use. For more information on encoding, see “Encoding for NLS” in the [SAS 9.2 National Language Support \(NLS\): Reference Guide](#).

### Details

The value for ENCODING= indicates that a different encoding will be used for the current session encoding. When you read data from a data set, SAS transcodes the data from the specified encoding to the session encoding. When you write data to a data set, SAS transcodes the data from the session encoding to the specified encoding.

### Input Processing

By default, encoding for input processing is determined as follows:

- If the session encoding and the encoding that is specified in the file are different, SAS transcodes the data to the session encoding.
- If a file has no encoding specified, but the file's data representation is different from the encoding of the current session, then SAS transcodes the data to the current session.

### Output Processing

By default, encoding for output processing is determined as follows:

- Data is written to a file using the encoding of the current session, except when a different output representation is specified using the OUTREP= data set option, the OUTENCODING= option in the LIBNAME statement, or the ENCODING= data set option.
- If a new file replaces an existing file, then the new file inherits the encoding of the existing file.
- If an existing file is replaced by a new file that was created under a different operating environment or that has no encoding specified, the new file uses the encoding of the current session.



**Note:** Character metadata and data output appears garbled if you specify a different encoding from where the data set was created.

In the following example, the data set to be printed is internally encoded as ASCII, however the data set option specifies an EBCDIC encoding. SAS attempts to transcode the data from EBCDIC to ASCII, but the data is already in ASCII. The result is garbled data.

```
data a;
x=1;
abc='abc';
run;
proc print data=a (encoding="ebcdic");
run;
```



**Note:** The following values for ENCODING= are invalid:

- UCS2
- UCS4
- UTF16
- UTF32

For a list of valid encoding values, see “Encoding Values for a SAS Session” in the [SAS 9.2 National Language Support \(NLS\): Reference Guide](#).

## IDXNAME= Table Option

Directs the Driver for BASE to use a specific index to match the conditions of a WHERE clause. Use with input data sets only. This option is mutually exclusive with [IDXWHERE= option](#).

### Syntax

```
IDXNAME=index-name
```

### Syntax Description

- **index-name**— Specifies the name (up to 32 characters) of a simple or composite index for the SAS data set. The Driver for BASE does not attempt to determine whether the specified index is the best one or whether a sequential search might be more resource efficient.

The specification is not a permanent attribute of the data set and is valid only for the current use of the data set.

### Details

By default, to satisfy the conditions of a WHERE clause for an indexed SAS data set, the Driver for BASE identifies zero or more candidate indexes that could be used to optimize the WHERE clause. From the list of candidate indexes, the Driver for BASE selects the one that it determines will provide the best performance, or rejects all of the indexes if a sequential pass of the data is expected to be more efficient.

Because the index the Driver for BASE selects might not always provide the best optimization, you can direct the Driver for BASE to use one of the candidate indexes by specifying the IDXNAME= table option. If you specify an index that the Driver for BASE does not identify as a candidate index, then IDXNAME= table option does not process the request; that is, IDXNAME= does not allow you to specify an index that would produce incorrect results.



## Comparisons

The `IDXWHERE=` table option enables you to override the Driver for BASE decision about whether to use an index.

## Examples

This example uses the `IDXNAME=` table option to direct the Driver for BASE to use a specific index to optimize the `WHERE` clause. The Driver for BASE then disregards the possibility that a sequential search of the data set might be more resource efficient and does not attempt to determine whether the specified index is the best one. (Note that the `EMPNUM` index was not created with the `NOMISS` option.)

```
create table mydata.empnew
as select * from mydata.employee {option idxname=empnum}
where empnum < 2000;
```

## IDXWHERE= Table Option

Specifies whether the Driver for BASE uses an index search or a sequential search to match the conditions of a `WHERE` clause. Use with input data sets only. This option is mutually exclusive with [IDXNAME= option](#).

## Syntax

```
IDXWHERE=YES|NO
```

## Syntax Description

- **YES** — Tells the Driver for BASE to choose the best index to optimize a `WHERE` clause, and to disregard the possibility that a sequential search of the data set might be more resource-efficient.
- **NO** — Tells the Driver for BASE to ignore all indexes and satisfy the conditions of a `WHERE` clause with a sequential search of the data set.



**Note:** You cannot use the `IDXWHERE=` table option to override the use of an index to process a `BY` statement.

## Details

By default, to satisfy the conditions of a `WHERE` clause for an indexed SAS data set, the Driver for BASE decides whether to use an index or to read the data set sequentially. The software estimates the relative efficiency and chooses the method that is more efficient.

You might need to override the software's decision by specifying the `IDXWHERE=` table option because the decision is based on general rules that occasionally might not produce the best results. That is, by specifying the `IDXWHERE=` table option, you are able to determine the processing method.



**Note:** The specification is not a permanent attribute of the data set and is valid only for the current use of the data set.

## Comparisons

The `IDXNAME=` table option enables you to direct the Driver for BASE to use a specific index.

## Examples

### Example 1: Specifying Index Usage

This example uses the `IDXWHERE=` table option to tell the Driver for BASE to decide which index is the best to optimize the `WHERE` clause. The Driver for BASE then disregards the possibility that a sequential search of the data set might be more resource-efficient:

```
create table mydata.empnew
  as select * from mydata.employee {option idxwhere=yes}
  where empnum < 2000;
```

### Example 2: Specifying No Index Usage

This examples uses the `IDXWHERE=` table option to tell the Driver for BASE to ignore any index and to satisfy the conditions of the `WHERE` clause with a sequential search of the data set:

```
create table mydata.empnew
  as select * from mydata.employee {option idxwhere=no}
  where empnum < 2000;
```

## LABEL= Table Option

Specifies a label for a Driver for BASE table.

### Syntax

```
LABEL='label'
```

### Syntax Description

- **'label'** — Specifies a text string of up to 256 characters. If the text contains single quotation marks, use double quotation marks around the label, or use two single quotation marks in the label text and surround the string with single quotation marks. To remove a label from a data set, assign a label that is equal to a blank that is enclosed in quotation marks.

### Details

You can use the `LABEL=` table option on both input and output tables. When you use the `LABEL=` table option on input tables, it assigns a label for the file for the duration of the operation. When this is specified for an output table, the label becomes a permanent part of that file.

A label assigned to a table remains associated with that table when you update a table in place, such as when you use the `APPEND` procedure or the `MODIFY` statement. However, a label is lost if you use a table with a previously assigned label to create a new table with the `CREATE TABLE` statement. For example, a label previously assigned to table `ONE` is lost when you create the new output table `ONE` in this `CREATE TABLE` statement:

```
create table one as select * from one;
```

## Examples

These examples assign labels to SAS data sets:

```
create table w2 {option label='1976 W2 Info, Hourly'};
create table new {option label='Peter's List'};
```

```
create table new {option label="Hillside's Daily Account"};
create table sales {option label='Sales For May(NE) '};
```

## LOCKTABLE= Table Option

Places shared or exclusive locks on tables.

### Syntax

```
LOCKTABLE=SHARE | EXCLUSIVE
```

### Syntax Description

- **SHARE** — Locks a table in shared mode, allowing other users or processes to read data from the tables, but preventing users from updating data.
- **EXCLUSIVE** — Locks a table exclusively, preventing other users from accessing any table that you open.

### Details

You can lock tables only if you are the owner or have been granted the necessary privilege.

If you access the Driver for BASE through the Federation Server, the default value for the LOCKTABLE option is SHARE.

## PW= Table Option

Assigns a READ, WRITE, and ALTER password to a SAS data set, and enables access to a password-protected file.

### Syntax

```
PW=password
```

### Syntax Description


- **password** — Specifies a valid Driver for BASE name.

### Details

The PW= table option applies only to a SAS data set. You can use this option to assign a password or to access a password-protected file. When a data set that is protected by a password is replaced, the new data set inherits the password.

When the code is written to the SAS log the password is blotted out, for example:

```
from a(pw=XXXXXXX);
```

 **Note:** A SAS password does not control access to a SAS file beyond the Driver for BASE or the SAS system. You should use the operating system-supplied utilities and file-system security controls to control access to SAS files outside of the Driver for BASE.

## READ= Table Option

Assigns a READ password to a SAS data set that prevents users from reading the file, unless they enter the password.

## Syntax

```
READ=read-password
```

## Syntax Description

- **read-password** — Specifies a valid Driver for BASE name.

## Details

The READ= option applies only to a SAS data set. You can use this option to assign a password or to access a read-protected file. When the code is written to the SAS log, the password is blotted out, for example:

```
from a(read=XXXXXXX);
```



**Note:** A SAS password does not control access to a SAS file beyond the Driver for BASE or the SAS system. You should use the operating system-supplied utilities and file-system security controls to control access to SAS files outside of the Driver for BASE.

## REUSE= Table Option

Specifies whether new rows can be written to freed space in a compressed SAS data set. Use with output data sets only.

## Syntax

```
REUSE=NO|YES
```

## Syntax Description

- **NO** — Does not track and reuse space in a compressed SAS data set. New rows are appended to the existing data set. Specifying NO results in less efficient data storage if you delete or update many rows in the data set.
- **YES** — Tracks and reuses space in a compressed SAS data set. New rows are inserted in the space that is freed when other rows are updated or deleted.

If you plan to use operations that add rows to the end of a compressed data set, use REUSE=NO. REUSE=YES causes new rows to be added wherever there is space in the file, not necessarily at the end of the file.

## Details

By default, new rows are appended to an existing compressed data set. If you want to track and reuse free space by deleting or updating other rows, use the REUSE= table option when you create a compressed SAS data set.

The REUSE= table option has meaning only when you are creating a new data set with the COMPRESS=YES option. Using the REUSE= table option when you are accessing an existing SAS data set has no effect.

## TYPE= Table Option

Specifies the data set type for a specially structured SAS data set.

## Syntax

```
TYPE=data-set-type
```

## Syntax Description

- **data-set-type** — Specifies the special type of the data set.

## Details

Use the TYPE= table option to create a special data set in the proper format or to identify the special type of the data set.

Most data sets do not have a specified type. However, there are several specially structured data sets that are used by some SAS/STAT procedures. These SAS data sets contain special columns and rows, and they are usually created by SAS statistical procedures. Because most of the special data sets are used with SAS/STAT software, they are described in the [SAS/STAT User's Guide](#).

Other values are available in other SAS software products and are described in the appropriate documentation.

## WRITE= Table Option

Assigns a WRITE password to a SAS data set that prevents users from writing to the file or that enables access to a write-protected file.

## Syntax

```
WRITE=write-password
```

## Syntax Description

- **write-password** — Specifies a valid Driver for BASE name.

## Details

The WRITE= table option applies only to a SAS data set. You can use this option to assign a password or to access a write-protected file. When the code is written to the SAS log, the password is blotted out, for example:

```
from a(write=XXXXXXX);
```



**Note:** A SAS password does not control access to a SAS file beyond the Driver for BASE or the SAS system. You should use the operating system-supplied utilities and file-system security controls to control access to SAS files outside of the Driver for BASE.

# SAP Reference

This section provides functionality details and guidelines for SAP® that are supported by the DataFlux® Driver for SAP (Driver for SAP).

The following topics provide additional information about the Driver for SAP:

- [Understanding the Driver for SAP](#)
- [Data Service Connection Arguments for SAP](#)
- [Data Types for SAP](#)

## Understanding the Driver for SAP

The Driver for SAP enables the Federation Server to read tables from SAP systems. The Driver for SAP has read-only capabilities.

The Driver for SAP supports most of the TSSQL functionality. The driver does not support the application's ability to submit native SQL statements.

The Driver for SAP is a remote driver, which means that it connects to a server process in order to access data. The process might be running on the same machine as the Federation Server, or it might be running on another machine in the network.

## Data Service Connection Arguments for SAP

To access data that is hosted on the Federation Server, a client must submit a connection string, which defines how to connect to the data.

The data service connection arguments for SAP include the following:

- **ABAPFM = abap\_function\_name** — Specifies the name of the Advanced Business Application Programming (ABAP) function module that the driver uses internally.

**Syntax:** APBAPFM | ABAPFUNCTION | ABAPFUNC = abap\_function\_name

**Default value:** /SAS/Z\_SAS\_DIALOG

- **ABAP\_NAMESPACE = namespace** — Specifies the namespace for ABAP functions and programs that are used by the driver. If the ABAP programs are installed in the customer namespace rather than in the default namespace, this parameter identifies where the ABAP programs are installed.

**Syntax:** ABAP\_NAMESPACE | ABAPNAMESPACE | ABAP\_NAME\_SPACE | ABAPNS | ABAP\_NS = namespace

**Default value:** /SAS/

- **ABAPPROG = abap\_program** — Specifies the name of the ABAP language that the driver uses internally. This value is set by the ABAP function module.

**Syntax:** ABAPPROG | ABAPREPORT | ABAPPROGRAM = abap\_program

**Default value:** /SAS/Z\_SAS\_READ

- **ASHOST = application\_server\_host** — Specifies the host name of the server or IP address of a specific application server.

**Syntax:** ASHOST | HST | RFCHOST | R3HOST = application\_server\_host

**Default value:** none

- **BATCH = 0 | 1 | Y | N** — Specifies whether the Driver for SAP should use SAP batch jobs for the data extracts.

- **Y** — The Driver for SAP uses batch jobs to extract R/3 data.
- **N** — The Driver for SAP uses dialog processes to extract R/3 data.

**Syntax:** BATCH | BATCH\_MODE | BATCHMODE = 0 | 1 | Y | N

**Default value:** N

- **BUFFER\_SIZE = buffersize** — Sets the minimum buffer size for data transfers in batch and dialog modes. The number of bytes should be greater than 10,000 and no more than eight digits.

**Syntax:** BUFFERSIZE | BUFFSIZE | BLOCK\_SIZE | BLOCKSIZE | BUFFER\_SIZE = buffersize

**Default value:** 100,000 bytes

- **CLIENT = client** — Specifies the SAP logon parameter client. Examples for a client are 000 or 800.



**Note:** When you access the SAP system via the driver, specify valid logon information including client, user name, password, and language. The user ID and password might also be provided through single sign-on (SSO). The driver performs a logon check at OPEN time.

**Syntax:** CLIENT | CLI | RFCCLIENT | RFCCLI = client

**Default value:** SAP system default

- **DESTGROUP = destination\_group** — Specifies the name of the destination group for batch access to the SAP system. The destination groups are defined in the /SAS/DESTS table in SAP.

**Syntax:** DESTGROUP = destination\_group

**Default value:** SAS1

- **DESTINATION = destination** — Specifies the destination in the `saprfc.ini` file, if working with a `saprfc.ini` file. If the RFC server is an R/3 system, this destination must also be defined in the `SIDEINFO` file for the SAP gateway.

**Syntax:** DESTINATION | DEST | DST | DSTN = destination

**Default value:** none

- **GROUP = application\_server\_group** — Specifies the name of the group of application servers, if load balancing is being used.

**Syntax:** GROUP = application\_server\_group

**Default value:** none

- **GWHOST = gateway\_host\_name** — Specifies the host name of the SAP gateway, if the server is R/2 or external.

**Syntax:** GWHOST | GATEWAY\_HOST = gateway\_host\_name

**Default value:** none

- **GWSERV = gateway\_service** — Specifies the service of the SAP gateway, if the server is an R/2 server or external.

**Syntax:** GWSERV | GATEWAY\_SERVICE = gateway\_service

**Default value:** none

- **IEEE\_REVERSE = Y | N** — Specifies whether floating point numbers are byte reversed.
  - **Y** — Specifies that floating-point numbers are byte reversed.
  - **N** — Specifies that floating-point numbers are not byte reversed.

**Syntax:** IEEE\_REVERSE = Y | N

**Default value:** For an R/3 application server on NT, the value is Y. The value is N for the other platforms.

- **INENCODING = code\_page** — Specifies the code page. Indicates the code page of the SAP server. The encoding is determined by the value returned by the SAP server. In some rare cases, it might be necessary to override this value by setting the inencoding = connection parameter.

**Syntax:** INENCODING = code\_page

**Default value:** none

- **LANGUAGE = language** — Specifies the SAP logon parameter language. The value for language is either the 2-byte ISO-language key or the 1-byte SAP-language. Examples for the language are EN, DE or E, D.



**Note:** When you access the SAP system via the driver, specify valid logon information including client, user name, password and language. The user ID and password might also be provided through SSO. The driver performs a logon check at OPEN time.

**Syntax:** LANGUAGE | LANG | LNG | RFCLANG | RFCLNG = language

**Default value:** SAP system default

- **MAX\_TABLE\_JOINS = number** — Specifies the number of tables that can be used in a left outer join or an inner join in ABAP Open SQL.

**Syntax:** MAX\_TABLE\_JOINS | MAX\_TABLES\_JOIN | MAX\_TABLES\_JOINS | MAX\_TABLE\_JOIN = number



**Default value:** 25

- **MSHOST = message\_server\_host** — Specifies the host name of the Message Server, if load balancing is being used.

**Syntax:** MSHOST= message\_server\_host

**Default value:** none

- **PWD = password** — Specifies the SAP logon parameter password.



**Note:** When you access the SAP system via the driver, specify valid logon information including client, user name, password and language. The user ID and password might also be provided through SSO. The driver performs a logon check at OPEN time.

**Syntax:** PASSWORD | PASSWD | PWD | PW | PASS = password

**Default value:** none

- **R3NAME = system\_name** — Specifies the name of the R/3 system, if load balancing is being used.

**Syntax:** R3NAME = system\_name

**Default value:** none

- **RFC\_STRING = additional\_rfc\_options** — Specifies additional logon or connection parameters for the RfcOpenEx() call. The Driver for SAP uses the RfcOpenEx() call to logon to the SAP system. Using this option, parameters which are not Driver for SAP connection attributes can be passed to the RfcOpenEx call.

For example:

```
RFC_STRING = "ABAP_DEBUG=1"
```

This option can be used to support future extensions of the RfcOpenEx call.

**Syntax:** RFC\_STRING | RFCSTRING | RFC\_OPTIONS\_EXT | RFCOPENEX |  
ADDITIONAL\_RFC\_OPTIONS = additional\_rfc\_options

**Default value:** none

- **SAPLOGON\_ID = saplogon\_id** — Specifies the string defined for SAPLOGON on a Windows® 32-bit system.

**Syntax:** SAPLOGON\_ID = saplogon\_id

**Default value:** none

- **SYSNR = system\_number** — Specifies the SAP system number, if load balancing is not being used. The number is the 2-byte code that identifies the system on the host, for example, 00 and 01.

**Syntax:** SYSNR |SYS |SYSTEM |SYSNO = system\_number

**Default value:** none

- **TRACE = 0 | 1 | Y | N** — Specifies whether or not the Driver for SAP should trace requests. If the trace option is switched on, the driver writes log information into a file. The RFC library logs messages in the `dev_rfc` file.
  - **0 or N** — RFC trace is switched off.
  - **1 or Y** — RFC trace is switched on.



**Note:** The RFC trace directory is set in the `RFC_TRACE_DIR` environment variable.

**Syntax:** `TRACE = 0 | 1 | Y | N`

**Default value:** 0

- **UID = user** — Specifies the SAP logon parameter user.




**Note:** When you access the SAP system via the driver, specify valid logon information including client, user name, password, and language. The user ID and password might also be provided through SSO. The driver performs a logon check at OPEN time.

**Syntax:** `USER | USR | RFCUSER | USERNAME | USERID = user`

**Default value:** none

## Data Types for SAP

The following table lists the SAP ABAP data types that are supported:

ABAP Data Type	Data Service Type Definition
ACCP	Specifies a posting period. The length is six digits and the format is YYYY.MM, for both input and output.
CHAR	Specifies a character string. In tables, the maximum field length is 255 characters. If longer character strings are used in tables, use the <a href="#">LCHR</a> data type.   <b>Note:</b> There are no length restrictions for character strings in structures.
CLNT	Specifies a client. Client fields have a length of three.
CUKY	Specifies a currency key. Field is referenced by type CURR. The length is five.
CURR	Specifies a currency field. Corresponds to the DEC field. Field refers to a field of type CUKY (reference field).
DATS	Specifies a date. The length is eight. The output mask can be defined with the user profile.
DEC	Specifies a counter or amount field with decimal point, sign, and comma. The maximum length is 31.
FLTP	Specifies a floating point number. The length, including decimals, is 16.

<b>ABAP Data Type</b>	<b>Data Service Type Definition</b>
INT1	Specifies a 1-byte integer, with a value range from 0 to 255.
INT2	Specifies a 2-byte integer, with a value range from -32767 to 32767. This is a length field and should precede the <a href="#">LCHR</a> and <a href="#">LRAW</a> data types. If an INSERT or UPDATE is made to the length field, the database interface enters the length actually used in the length field.
INT4	Specifies a 4-byte integer, with a value range from 2177483647 to 2177483647.
LANG	<p>Specifies a language key, which has its own field format for special functions. The language key is displayed with a length of two in the Federation Server Manager, but is stored with a length of one in the database.</p> <p>The conversion exit, ISOLA, converts the value to be displayed in the Federation Server Manager to that of the database and the opposite is true. This conversion exit is automatically assigned to a domain of this type during activation.</p>
LCHR	Specifies a character string of at least 256 characters. Fields must be at the end of transparent tables and have a preceding length field of type <a href="#">INT2</a> . For an INSERT or UPDATE in ABAP programs, specify the actual length that is required. Fields of this type cannot be used in the WHERE condition of a SELECT statement.
LRAW	Specifies an uninterpreted string of at least 256 bytes. Fields must be at the end of transparent tables and have a preceding length field of type <a href="#">INT2</a> . For an INSERT or UPDATE in ABAP programs, specify the actual length that is required.
NUMC	Specifies a character string that contains digits only. The maximum length is 255.
PREC	Specifies the precision of a <a href="#">QUAN</a> field. The length is two.
QUAN	Specifies a quantity which corresponds to the <a href="#">DEC</a> field. A field of this type must always refer to a units field with format UNIT (reference field).
RAW	Specifies an uninterpreted byte string, with a maximum length of 255 in tables. If longer RAW fields are used in tables, you the <a href="#">LRAW</a> type. This length restriction is not valid for structures.
TIMS	Specifies a time value in the format <i>hh.mm.ss</i> .
UNIT	Specifies a units key and referenced by fields of type <a href="#">QUAN</a> . The length is two or three.
VARC	Specifies a character string of variable length. As of SAP release 3.0, creating fields of this data type is no longer supported. Existing fields with this data type can still be used. Fields of this type cannot be used in the WHERE condition of a SELECT statement.

# DB2 Reference

This section provides functionality details and guidelines for DB2® that are supported by the DataFlux® Driver for DB2 (Driver for DB2) under UNIX® and PC hosts.

The following topics provide additional information about the Driver for DB2:

- [Understanding the Driver for DB2](#)
- [Database Functionality Support for DB2](#)
- [Data Service Connection Arguments for DB2](#)
- [Data Types for DB2](#)
- [Load Data by Using Bulk Operations for DB2](#)
- [Naming Conventions for DB2](#)
- [TSSQL Statement Table Options](#)

## Understanding the Driver for DB2

The Driver for DB2 enables the Federation Server to read and update legacy DB2 tables. In addition, the driver creates DB2 tables that can be accessed by both the Federation Server and the DB2 database management system (DBMS).

The Driver for DB2 supports most of the TSSQL functionality. The driver also supports an application's ability to submit native DB2 SQL statements.

The Driver for DB2 is a remote driver, which means that it connects to a server process in order to access data. The process might be running on the same machine as the Federation Server, or it might be running on another machine in the network.

## Database Functionality Support for DB2

The following table lists the database functionality support level for DB2 that is provided by the Driver for DB2 under UNIX and PC hosts. For general information about database functionality, see [Database Functionality Supported by Data Services](#).

Functionality	Description	Support	Comment
bookmark	Stored location used for quick retrieval and scrollable cursors		
bulk operations	Ability to add, delete, or update a large quantity of data with one operation	Supports additions, deletions, and updates	For more information, see <a href="#">Load Data by Using Bulk Operations for DB2 under UNIX and PC Hosts</a> .
cursor support	Functionality that keeps track of the position in a result set and enables	Forward-only, static, and keyset cursors	Cursor functionality is supported by the driver or by specifying that the

Functionality	Description	Support	Comment
	multiple operations on the result set		Federation Server provides additional cursor support.
index	Component of a database that contains information about the physical location of data stored in a file	Supported	
integrity constraint	Functionality that ensures correctness, completeness, and compliance of data	Supported	
multi-row fetch	Enables an application to request more than one row at a time	Supported	
stored routines	SQL programs that perform computations or other tasks consistently	Result set stored procedures with parameters	
threaded I/O processing	Efficient method of processing a query with multiple threads	Supported if result set contains an integer type of column that can be used by the SQL MOD() scalar function	
transactions	Process of protecting data by ensuring the integrity of data during multi-user update sessions	Supported	

## Data Service Connection Arguments for DB2

To access data that is hosted on the Federation Server, a client must submit a connection string, which defines how to connect to the data. The data service connection arguments for a DB2 data set include connection options and advanced options.

### Connection Options

Connection options are used to establish a connection to a data service. For each connection to the specified data service, one or more connection options should be specified in the create data service data definition language (DDL) statement. For more information on this DDL statement, see [Create Data Service](#).

The following connection options are supported for DB2 data sets:

- **CATALOG=catalog-identifier;** — Specifies an arbitrary identifier for an SQL catalog, which groups logically related schemas. Any identifier is valid (for example, catalog=db2).

**Requirement:** You must specify a catalog. For the DB2 database, this is a logical catalog name to use as an SQL catalog identifier.

- **DATABASE|DB=database-specification;** — Specifies the name of the DB2 database. For example, database=sample.

**Requirement:** You must specify a database name.

- **DRIVER=DB2;** — Identifies the data service to which you want to connect, which is a DB2 database.

**Requirement:** You must specify the driver.

## Advanced Options

Advanced driver options are additional options that are not required in order to connect to the data source. They are used to establish connections to catalogs, data source names (DSNs), and schemas. While advanced options can also be used when connecting to a data service, doing so will cause the specified options to apply to all data service connections.

The following advanced options are supported for DB2 data sets:

- **CLIENT\_ENCODING=encoding-value;** — Specifies the encoding of the DB2CODEPAGE to the DB2 data service driver.

When the encoding of the DB2 client layer (stored in DBCODEPAGE) is different from the encoding value of the DB2 operating system value, which is generally the SAS® session encoding value, the DB2 client layer attempts to convert incoming data to the DB2 encoding value that is stored in DB2CODEPAGE. You must first determine the correct value for DB2CODEPAGE and then set the CLIENT\_ENCODING= option to match the corresponding encoding value in DB2CODEPAGE in order to prevent the client layer from converting the data incorrectly.

For example, suppose you are storing Japanese characters in a DB2 database and the client machine where the DB2 data service driver is executing is a Windows® machine running CP1252 encoding. When the application tries to extract the data into the Federation Server, the DB2 client layer attempts to convert these Japanese characters into Latin1 representation, which does not contain Japanese characters. As a result, a garbage character appears to indicate a failure in transcoding.

To resolve this situation, you must first set the DB2CODEPAGE value to 1208 (the IBM® code page value that matches UTF-8 encoding) to specify that the DB2 client layer send the data to the application in UTF-8 instead of converting it into Latin1. In addition, you must specify to the DB2 data service driver the corresponding encoding value of DB2CODEPAGE because the Driver for DB2 cannot derive this information from a DB2 session. For this particular Windows case, set the CLIENT\_ENCODING= option to the UTF-8 to match the DB2CODEPAGE value (1208) in order to specify the DB2CODEPAGE value to the DB2 data service driver.

However, changing the value of DB2CODEPAGE affects all applications that run on that machine. You should reset the value to the usual DB2CODEPAGE value, which was derived when the database was created.



**Caution:** Setting the DB2CODEPAGE value or the CLIENT\_ENCODING= value incorrectly can cause unpredictable results.

You should set these values only when a situation such as the example above occurs.



**Note:** You can specify any valid encoding value for `CLIENT_ENCODING=option`.

- **DRIVER\_TRACE='API|SQL|DRIVER|ALL'** — Requests tracing information, which logs transaction records to an external file that can be used for debugging purposes. The Federation Server driver writes a record of each command that is sent to the DBMS to the trace log based on the specified tracing level, which determines the type of tracing information. The tracing levels are as follows:
  - **API** — Specifies that API method calls be sent to the trace log. This option is most useful if you are having a problem and need to send a trace log to DataFlux Technical Support for troubleshooting.
  - **SQL** — Specifies that SQL statements that are sent to the DBMS be sent to the trace log. Tracing information is DBMS specific, but most Federation Server drivers log SQL statements such as `SELECT` and `COMMIT`.
  - **DRIVER** — Specifies that driver-specific information be sent to the trace log.
  - **ALL** — Activates all trace levels.

**Default:** Tracing is not activated.

**Requirement:** If you activate tracing, you must also specify the location of the trace log with `DRIVER_TRACEFILE=`. (Optional) You can control trace log formatting with `DRIVER_TRACEOPTIONS=`.

**Interaction:** You can specify one trace level, or you can concatenate more than one by including the `|` (OR) symbol. For example: `driver_trace='api|sql'` specifies to generate tracing information for API calls and SQL statements.

- **DRIVER\_TRACEFILE='filename';** — Specifies where to write trace information. The trace log is a text file. Include the filename and extension in single or double quotation marks. For example:

```
driver_tracefile='\\mytrace.log'
```

**Default:** There is no default trace log location.

**Requirement:** You must specify a location for a trace log if you activate tracing with `DRIVER_TRACE=`.

**Interaction:** (Optional) You can control trace log formatting with `DRIVER_TRACEOPTIONS=`.

- **DRIVER\_TRACEOPTIONS=APPEND|THREADSTAMP|TIMESTAMP;** — Specifies options to control formatting and other properties for the trace log:
  - **APPEND** — Adds trace information to the end of an existing trace log. The contents of the file are not overwritten.
  - **THREADSTAMP** — Prepends each line of the trace log with a thread identification.
  - **TIMESTAMP** — Prepends each line of the trace log with a time stamp.

**Default:** The trace log is overwritten with no thread identification or time stamp.

- **PWD=password;** — Specifies the DB2 password.
- **UID=userid;** — Specifies the DB2 login user ID.

## Data Types for DB2

When an application submits Federation Server language statements, only the Federation Server data types are supported for column definition. Native DB2 data types that are not supported by Federation Server are mapped to similar Federation Server data types for data retrieval. For information about the Federation Server data types, see [Data Service Data Types](#).

When an application submits DB2 SQL statements, native DB2 data types are supported. Submitting DB2 SQL statements is available only to applications that consume JDBC, ODBC, or OLE DB client-side drivers. For more information about the native DB2 SQL data types, see your IBM documentation.

The following table lists the data types that are supported for DB2:

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
BIGINT	BIGINT	Specifies a large signed, exact whole number with a precision of 19.	BIGINT
*	BLOB( <i>n</i> [K M G] )	Specifies a varying-length binary large object string of up to 2 gigabytes (2,147,483,647 bytes). Maximum length specified in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes).	*
CHAR( <i>n</i> )	CHAR( <i>n</i> )	Specifies a fixed-length character string from 1 to 254 characters; the default is 1.	CHAR
*	CHAR( <i>n</i> ) FOR BIT DATA	Specifies a fixed-length binary string from 1 to 254 characters; the default is 1.	*
*	CLOB( <i>n</i> [K M G] )	Specifies a varying-length character large object string up to 2 gigabytes (2,147,483,647 bytes). Maximum length specified in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes).	*
DATE	DATE	Specifies a date value in the range 01-01-0001 to 12-31-9999 in the format	DATE



Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
		<i>mm-dd-yyyy</i> (United States) and <i>dd.mm.yyyy</i> (European standard format).	
*	DBCLOB ( <i>n</i> [K M G])	Specifies a varying-length double-byte character large object string of up to 1 gigabyte (1,073,741,823 bytes). Maximum length specified in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes).	*
*	DECIMAL( <i>p,s</i> )	Specifies a signed, fixed-point decimal number. Precision range is 1 to 31; the default is 5. Scale range is 1 to 31; the default is 0. Precision can be greater than what is specified; scale must be less than or equal to precision.	*
DOUBLE	DOUBLE	Specifies a signed, double precision, floating-point number with a precision of 53.	DOUBLE
FLOAT( <i>p</i> )	FLOAT( <i>p</i> )	Specifies a signed, single precision or double precision, floating-point number with a precision range of 1 to 24 (single precision) or 25 to 53 (double precision); the default is 53.	FLOAT
*	GRAPHIC( <i>n</i> )	Specifies a fixed-length graphic string from 1 to 127 double-byte characters; the default is 1.	*
INTEGER	INTEGER	Specifies a regular signed, exact whole number with a precision of 10.	INTEGER
*	LONG VARCHAR [FOR BIT DATA]	Specifies a varying-length character or binary string from 1 to 32,700 characters.	*
*	LONG VARGRAPHIC( <i>n</i> )	Specifies a varying-length graphic string from 1 to 32,700 double-byte characters.	*
*	NUMERIC( <i>p,s</i> )	Specifies a signed, fixed-point decimal number. Precision range is 1 to 31; the default is 5. Scale range is 1 to 31; the default is 0. Scale must be less than or equal to precision.	*
REAL	REAL	Specifies a signed, single precision, floating-point number with a precision of 24.	REAL
SMALLINT	SMALLINT	Specifies a small signed, exact whole number with a precision of 5.	SMALLINT

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
TIME( <i>p</i> )	TIME( <i>p</i> )	Specifies a time value in the format <i>hh.mm.ss</i> (United States) and <i>hh.mm[.ss]</i> European standard.	TIME
TIMESTAMP( <i>p</i> )	TIMESTAMP( <i>p</i> )	Specifies a date, time, and optional microsecond values in the range of 00-01-0001, 00 hours, 00 minutes, 00 seconds, and 000000 microseconds to 12-31-9999, 24 hours, 00 minutes, 00 seconds, and 000000 microseconds in the format <i>yyyy-mm-dd-hh.mm.ss[.nnnnnn]</i> .	TIMESTAMP
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Specifies a varying-length character string from 1 to 4,000 characters.	VARCHAR
VARBINARY( <i>n</i> )	VARCHAR( <i>n</i> ) FOR BIT DATA	Specifies a varying-length binary string from 1 to 4,000 characters.	VARBINARY
*	VARGRAPHIC( <i>n</i> )	Specifies a varying-length graphic string from 1 to 2,000 double-byte characters.	*

\* For the Federation Server languages, the data type cannot be defined, and when data is retrieved, the Federation Server maps the native DB2 data type to a similar Federation Server data type.

## Load Data by Using Bulk Operations for DB2

Bulk loading is the fastest way to insert large numbers of rows into a DB2 table. This facility enables you to insert rows two to ten times more quickly than using regular SQL INSERT statements. To use the bulk load facility, specify the TSSQL table option BULKLOAD=YES.

The Driver for DB2 under UNIX and PC hosts supports the CLI LOAD bulk loading method, which is available only in DB2 Version 7 FixPak 4 and later clients and servers. To use CLI LOAD, you must have system administrator authority, database administrator authority, or load authority on the database and the insert privilege on the table that is being loaded.

The following are DB2 bulk-loading options that can be specified as data set options in TSSQL SELECT statements:

- BL\_LOG= to specify the path to a DB2 load log.
- BL\_LOAD\_REPLACE= to specify whether DB2 appends (NO) or replaces (YES) rows during bulk loading.
- BL\_OPTIONS= to pass DB2 file type modifiers to CLI LOAD for additional loading and processing specifications. See your DB2 documentation for the list of file type modifiers available for CLI LOAD.

To specify options in TSSQL, use curly braces:

```
insert into db2.db2table {options bulkload=yes
    bulkopts=(b1_log=drive:\temp\cliload.log b1_replace=yes)}
select * from work.sastable;
```

For more information about using options in TSSQL, see the *DataFlux Federation Server: TSSQL Language Reference*.

## Naming Conventions for DB2

DB2 is case insensitive. All names default to uppercase unless they are quoted.

DB2 objects include tables, views, columns, and indexes. Use the following naming conventions for them:

- A name can start with a letter or with one of the following symbols: the dollar sign (\$), the number (or pound) sign (#), or the at symbol (@).
- A name can be from 1 to 18 characters long.
- A name can contain the letters A through Z, any valid letter that has an accent (such as the letter a), the digits 0 through 9, the underscore (\_), the dollar sign (\$), the number or pound sign (#), or the at symbol (@).
- A name is not case sensitive (for example, the table name CUSTOMERS is the same as Customers), but object names are converted to uppercase when they are typed. If a name is enclosed in quotation marks, then the name is case sensitive.
- A name cannot be a DB2 or an SQL reserved word, such as WHERE or VIEW.
- A name cannot be the same as another DB2 object that has the same type.

Schema and database names have similar conventions, except that they are each limited to eight characters. For more information, see your DB2 SQL reference manual.

## TSSQL Statement Table Options

Federation Server TSSQL (TSSQL) statement table options specify actions that apply only to the table with which they appear. They enable you to perform operations such as specifying a password for the driver file and renaming columns.

You can specify table options on a TSSQL statement on which you specify a table name, such as the CREATE TABLE, ALTER TABLE, or SELECT statement.

For more information on how to use these options, see "Using TSSQL Statement Table Options" in the *DataFlux Federation Server TSSQL Reference Guide*.

### Syntax

Specify a TSSQL statement table option immediately after the table name, within curly braces { } and including the keyword OPTION. To specify several table options, separate them with spaces or commas.

```
{OPTION option-1=value [... option-n=value] }
```



**Important:** While specifying the syntax for table options, you cannot have a space between the left curly brace { and the keyword OPTION. A space will result in a syntax error.

These examples show table options in TSSQL statements:

```
create table salary {option encrypt=yes read=green};  
select * from salary {option read=green};
```

The following statement table options are supported by the Driver for DB2:

- [BL\\_ALLOW\\_READ\\_ACCESS= Table Option](#)
- [BL\\_COPY\\_LOCATION= Table Option](#)
- [BL\\_CPU\\_PARALLELISM= Table Option](#)
- [BL\\_DATA\\_BUFFER\\_SIZE= Table Option](#)
- [BL\\_DISK\\_PARALLELISM= Table Option](#)
- [BL\\_EXCEPTION= Table Option](#)
- [BL\\_INDEXING\\_MODE= Table Option](#)
- [BL\\_LOAD\\_REPLACE= Table Option](#)
- [BL\\_LOG= Table Option](#)
- [BL\\_OPTIONS= Table Option](#)
- [BL\\_RECOVERABLE= Table Option](#)
- [BL\\_REMOTE\\_FILE= Table Option](#)
- [BL\\_WARNING\\_COUNT= Table Option](#)
- [BULKLOAD= Table Option](#)
- [BULKOPTS= Table Option](#)

## **BL\_ALLOW\_READ\_ACCESS= Table Option**

Specifies that the original table data is still visible to readers during bulk load.

### **Syntax**

```
BL_ALLOW_READ_ACCESS=YES | NO
```

### **Syntax Description**

- **YES** — Specifies that the original (unchanged) data in the table is still visible to readers while bulk load is in progress.
- **NO** — Specifies that readers cannot view the original data in the table while bulk load is in progress.

## Details

To specify this option, you must first set BULKLOAD=YES.

For more information, see [BULKLOAD= Table Option](#).

## BL\_COPY\_LOCATION= Table Option

Specifies the directory to which DB2 saves a copy of the loaded data.

### Syntax

```
BL_COPY_LOCATION=pathname
```

### Syntax Description

- ***pathname*** — Specifies the path where the loaded data will be copied.

## Details

To specify this option, you must first set BULKLOAD=YES. This option is valid only when BL\_RECOVERABLE=YES.

For more information, see [BULKLOAD= Table Option](#) and [BL\\_RECOVERABLE= Table Option](#).

## BL\_CPU\_PARALLELISM= Table Option

Specifies the number of processes or threads to use when building table objects.

### Syntax

```
BL_CPU_PARALLELISM=number of processes or threads
```

### Syntax Description

- **number of processes or threads** — Specifies the number of processes or threads that the load utility uses to parse, convert, and format data records when building table objects.

## Details

To specify this option, you must first set BULKLOAD=YES.

This option exploits intrapartition parallelism and significantly improves load performance. It is particularly useful when loading presorted data, because record order in the source data is preserved.

The maximum number that is allowed is 30. If the value of this parameter is 0 or has not been specified, the load utility selects an intelligent default that is based on the number of available CPUs on the system at run time. If there is insufficient memory to support the specified value, the utility adjusts the value.

When BL\_CPU\_PARALLELISM is greater than one, the flushing operations are asynchronous, permitting the loader to exploit the CPU. If tables include either LOB or LONG VARCHAR

data, parallelism is not supported and this option is set to one, regardless of the number of system CPUs or the value that the user specified.

Although use of this parameter is not restricted to symmetric multiprocessor (SMP) hardware, you might not obtain any discernible performance benefit from using it in non-SMP environments.

For more information about using `BL_CPU_PARALLELISM=`, see the `CPU_PARALLELISM` parameter in the *IBM DB2 Universal Database Data Movement Utilities Guide and Reference*. Also, see [BL\\_DATA\\_BUFFER\\_SIZE= Table Option](#), [BL\\_DISK\\_PARALLELISM= Table Option](#), and [BULKLOAD= Table Option](#).

## BL\_DATA\_BUFFER\_SIZE= Table Option

Specifies the total amount of memory to allocate for the bulk load utility to use as a buffer for transferring data.

### Syntax

```
BL_DATA_BUFFER_SIZE=buffer-size
```

### Syntax Description

- **buffer-size** — Specifies the total amount of memory (in 4KB pages), regardless of the degree of parallelism, that is allocated for the bulk load utility to use as buffered space for transferring data within the utility.

### Details

To specify this option, you must first set `BULKLOAD=YES`.

If you specify a value that is less than the algorithmic minimum, the minimum required resource is used and no warning is returned. This memory is allocated directly from the utility heap, the size of which you can modify through the `util_heap_sz` database configuration parameter. If you do not specify a value, the utility calculates an intelligent default at run time that is based on a percentage of the free space that is available in the utility heap at the time of instantiation of the loader, as well as some characteristics of the table.

It is recommended that the buffer be several extents in size. An extent is the unit of movement for data within DB2, and the extent size can be one or more 4KB pages. The `DATA BUFFER` parameter is useful when you are working with large objects (LOBs) because it reduces I/O waiting time. The data buffer is allocated from the utility heap. Depending on the amount of storage available on your system, you should consider allocating more memory for use by the DB2 utilities. You can modify the database configuration parameter `util_heap_sz` accordingly. The default value for the Utility Heap Size configuration parameter is 5000 4KB pages. Because load is only one of several utilities that use memory from the utility heap, it is recommended that no more than 50 percent of the pages defined by this parameter be made available for the load utility, and that the utility heap be defined large enough.

For more information about using this option, see the `DATA BUFFER` parameter in the *IBM DB2 Universal Database Data Movement Utilities Guide and Reference*. Also, see [BL\\_CPU\\_PARALLELISM= Table Option](#), [BL\\_DISK\\_PARALLELISM= Table Option](#), and [BULKLOAD= Table Option](#).

## BL\_DISK\_PARALLELISM= Table Option

Specifies the number of processes or threads to use when writing data to disk.

### Syntax

```
BL_DISK_PARALLELISM=number of processes or threads
```

### Syntax Description

- **number of processes or threads** — Specifies the number of processes or threads that the load utility uses to write data records to the table-space containers.

### Details

To specify this option, you must first set BULKLOAD=YES.

This option exploits the available containers when it loads data and significantly improves load performance.

The maximum number that is allowed is the greater of four times the BL\_CPU\_PARALLELISM value, which the load utility actually uses, or 50. By default, BL\_DISK\_PARALLELISM is equal to the sum of the table-space containers on all table spaces that contain objects for the table that is being loaded except where this value exceeds the maximum number that is allowed.

If you do not specify a value, the utility selects an intelligent default that is based on the number of table-space containers and the characteristics of the table.

For more information about using this option, see the DISK\_PARALLELISM parameter in the *IBM DB2 Universal Database Data Movement Utilities Guide and Reference*. Also, see [BL\\_CPU\\_PARALLELISM= Table Option](#), [BL\\_DATA\\_BUFFER\\_SIZE= Table Option](#), and [BULKLOAD= Table Option](#).

## BL\_EXCEPTION= Table Option

Specifies the exception table into which rows in error are copied.

### Syntax

```
BL_EXCEPTION=exception table-name
```

### Syntax Description

- **exception table-name** — Specifies the exception table into which rows in error are copied.

### Details

To specify this option, you must first set BULKLOAD=YES.

Any row that is in violation of a unique index or a primary key index is copied. DATALINK exceptions are also captured in the exception table. If you specify an unqualified table name, the table is qualified with the CURRENT SCHEMA. Information that is written to the exception table is not written to the dump file. In a partitioned database environment, you

must define an exception table for those partitions on which the loading table is defined. However, the dump file contains rows that cannot be loaded because they are not valid or contain syntax errors.

For more information about using this option, see the FOR EXCEPTION parameter in the *IBM DB2 Universal Database Data Movement Utilities Guide and Reference*. For more information about the load exception table, see the load exception table topics in the *IBM DB2 Universal Database Data Movement Utilities Guide and Reference* and the *IBM DB2 Universal Database SQL Reference, Volume 1*. Also, see [BULKLOAD= Table Option](#).

## BL\_INDEXING\_MODE= Table Option

Specifies which scheme the DB2 load utility should use for index maintenance.

### Syntax

```
BL_INDEXING_MODE=AUTOSELECT | REBUILD | INCREMENTAL | DEFERRED
```

### Syntax Description

- **AUTOSELECT** — Specifies that the load utility automatically decides between REBUILD or INCREMENTAL mode.
- **REBUILD** — Specifies that all indexes are rebuilt.
- **INCREMENTAL** — Specifies that indexes are extended with new data.
- **DEFERRED** — Specifies that the load utility does not attempt index creation if this mode is specified. Indexes are marked as needing a refresh.

### Details

To specify this option, you must first set BULKLOAD=YES.

For more information about using the values for this option, see the *IBM DB2 Universal Database Data Movement Utilities Guide and Reference*. Also, see [BULKLOAD= Table Option](#).

## BL\_LOAD\_REPLACE= Table Option

Specifies whether DB2 will append or replace rows during bulk loading.

### Syntax

```
BL_LOAD_REPLACE=NO | YES
```

### Syntax Description

- **NO** — Specifies that the CLI LOAD interface appends new rows of data to the DB2 table. The default value.
- **YES** — Specifies that the CLI LOAD interface replaces the existing data in the table.

### Details

To use this table option you must first specify BULKLOAD=YES.



This table option is specified within the BULKOPTS= container option, for example:

```
bulkload=yes;  
bulkopts=(bl_load_replace=yes);
```

For more information, see [BULKOPTS= Table Option](#).

## BL\_LOG= Table Option

Identifies a log file that will contain information such as statistics and error information for a bulk load.

### Syntax

```
BL_LOG=path-and-log-filename
```

### Syntax Description

- ***path-and-log-filename*** — Specifies a file to which information about the loading process is written. The default value is DBMS-specific.

### Details

When the DBMS bulk load facility is invoked, it creates a log file. The contents of the log file are DBMS-specific. The BL\_ prefix distinguishes this log file from the one created by the SAS log. If the BL\_LOG= table option is specified with the same path and filename as an existing log, the new log replaces the existing log.

To specify this table option, you must first specify BULKLOAD=YES.

If the BL\_LOG= table option is not specified, the log file is deleted automatically after a successful operation.

This option is specified with the BULKOPTS= container option, for example:

```
bulkload=yes; bulkopts=(bl_log="c:\temp\bulkload.log");
```

For more information, see [BULKOPTS= Table Option](#).

## BL\_OPTIONS= Table Option

Passes options to the DBMS bulk load facility, affecting how it loads and processes data.

### Syntax

```
BL_OPTIONS='option [, ...option ] '
```

### Syntax Description

- ***option*** — Specifies a valid option.

### Details

To specify this option you must first specify BULKLOAD=YES.

The `BL_OPTIONS=` table option enables you to pass options to the DBMS bulk load facility when it is invoked, thereby affecting how data is loaded and processed. You must separate multiple options with commas and enclose the entire string of options in single quotation marks.

This option passes DB2 file-type modifiers to DB2 LOAD or IMPORT commands to affect how data is loaded and processed. Not all DB2 file type modifiers are appropriate for all situations. You can specify one or more DB2 file type modifiers with .IXF files. For a list of file type modifiers, see the description of the LOAD and IMPORT utilities in the *DB2 Data Movement Utilities Guide and Reference*.

This option is specified with the `BULKOPTS=` container option, for example:

```
bulkload=yes; bulkopts=(bl_options='option1, option2');
```

For more information, see [BULKOPTS= Table Option](#).

## BL\_RECOVERABLE= Table Option

Specifies whether the LOAD process is recoverable.

### Syntax

```
BL_RECOVERABLE=YES | NO
```

### Syntax Description

- **YES** — Specifies that the LOAD process is recoverable. YES also specifies that the copy location for the data should be specified by `BL_COPY_LOCATION=`.
- **NO** — Specifies that the LOAD process is not recoverable.

### Details

To specify this option, you must first set `BULKLOAD=YES`.

For more information, see [BULKLOAD= Table Option](#).

## BL\_REMOTE\_FILE= Table Option

Specifies the base filename and location of DB2 LOAD temporary files.

### Syntax

```
BL_REMOTE_FILE=pathname-and-base-filename
```

### Syntax Description

- **pathname-and-base-filename** — Specifies the full pathname and base filename to which DB2 appends extensions (such as .log, .msg, and .dat files) to create temporary files during load operations. By default, the base filename takes the form `BL_<table>_<unique-ID>`:
  - **table** — Specifies the table name.

- **unique-ID** — Specifies a number that is used to prevent collisions in the event of two or more simultaneous bulk loads of a particular table.

## Details

To specify this option, you must first set BULKLOAD=YES.

When you specify this option, the DB2 LOAD command is used (instead of the IMPORT command).

For *pathname*, specify a location on a DB2 server that is accessed exclusively by a single DB2 server instance, and for which the instance owner has read and write permissions. Make sure that each LOAD command is associated with a unique *pathname-and-base-filename* value.

For more information, see [BULKLOAD= Table Option](#).

## BL\_WARNING\_COUNT= Table Option

Specifies the maximum number of row warnings to allow before the load fails.

### Syntax

```
BL_WARNING_COUNT=warning-count
```

### Syntax Description

- **warning-count** — Specifies the maximum number of row warnings to allow before the load fails.

## Details

To specify this option, you must first set BULKLOAD=YES and also specify a value for BL\_REMOTE\_FILE=.

Use this option to limit the maximum number of rows that generate warnings. See the log file for information about why the rows generated warnings.

For more information, see [BL\\_REMOTE\\_FILE= Table Option](#) and [BULKLOAD= Table Option](#).

## BULKLOAD= Table Option

Loads rows of data as one unit.

### Syntax

```
BULKLOAD=YES | NO
```

### Syntax Description

- **YES** — Calls a DBMS-specific bulk load facility in order to insert or append rows to a DBMS table.
- **NO** — Does not call the DBMS-specific bulk load facility. This is the default value.

## Details

Using BULKLOAD=YES is the fastest way to insert rows into a DBMS table.

You can specify additional options. See [BL\\_LOAD\\_REPLACE= Table Option](#), [BL\\_LOG= Table Option](#), and [BL\\_OPTIONS= Table Option](#). These options require the BULKOPTS= container option. For more information, see [BULKOPTS= Table Option](#).

## BULKOPTS= Table Option

Container option for BL\_LOAD\_REPLACE=, BL\_LOG=, and BL\_OPTIONS= table options. The category is Bulk Loading. This option must follow BULKLOAD=YES. For more information, see [BULKLOAD= Table Option](#).

## Syntax

```
BULKOPTS= (option [, ...option ])
```

## Syntax Description

- **BL\_LOAD\_REPLACE=NO|YES** — Specifies whether the CLI LOAD interface replaces the existing data in the table. For more information, see [BL\\_LOAD\\_REPLACE= Table Option](#).
- **BL\_LOG=pathname** — Specifies a file to which information about the loading process is written. For more information, see [BL\\_LOG= Table Option](#).
- **BL\_OPTION=option** — Specifies a valid DB2 option. For more information, see [BL\\_OPTIONS= Table Option](#).

## Details

The BULKOPTS= option is a container option that is required in order to specify BL\_LOAD\_REPLACE=, BL\_LOG=, and BL\_OPTION= table options. To specify BULKOPTS= option you must first specify BULKLOAD=YES. For more information, see [BULKLOAD= Table Option](#).

## Examples

The following example uses BULKLOAD=YES and BULKOPTS= options:

```
BULKLOAD=YES;  
BULKOPTS=(BL_LOG='drive:\temp\bulkload.log' BL_LOAD_REPLACE=yes  
          BL_OPTIONS='ERRORS=999, LOAD=2000');
```

# MySQL Reference

This section provides functionality details and guidelines for the MySQL® server versions 5.0 and 5.1 that are supported by the DataFlux® Driver for MySQL (Driver for MySQL).

The following topics provide additional information about the Driver for MySQL:

- [Understanding the Driver for MySQL](#)
- [Database Functionality Support for MySQL](#)
- [Data Service Connection Arguments for MySQL](#)
- [Data Types for MySQL](#)
- [Naming Conventions for MySQL](#)
- [TSSQL Statement Table Options](#)

## Understanding the Driver for MySQL

The Driver for MySQL enables the Federation Server to read and update MySQL tables. In addition, the driver creates MySQL tables that can be accessed by both the Federation Server and the MySQL database.

The Driver for MySQL supports most of the TSSQL functionality. The driver also supports an application's ability to submit native MySQL SQL statements.

The Driver for MySQL is a remote driver, which means that it connects to a server process in order to access data. The process might be running on the same machine as the Federation Server, or it might be running on another machine in the network.

## Database Functionality Support for MySQL

The following table lists the database functionality support level for MySQL that is provided by the Driver for MySQL. For general information about database functionality, see [Database Functionality Supported by Data Services](#).

Functionality	Description	Support	Comment
bookmark	Stored location used for quick retrieval and scrollable cursors	Not Supported	
bulk operations	Ability to add, delete, or update a large quantity of data with one operation	Not supported	
cursor support	Functionality that keeps track of the position in a result set and enables multiple operations on the result set	Static cursor	Cursor functionality is supported by the driver or by specifying that the Federation Server provides additional cursor support.
index	Component of a database		

Functionality	Description	Support	Comment
	that contains information about the physical location of data stored in a file		
integrity constraint	Functionality that ensures correctness, completeness, and compliance of data	Supported	
locking	Process by which a database restricts access in a multi-user environment	Row-level (optimistic) and read locking	
multi-row fetch	Enables an application to request more than one row at a time	Supported	
stored routines	SQL programs that perform computations or other tasks consistently	Stored procedures with parameters	
threaded I/O processing	Efficient method of processing a query with multiple threads	Not supported	
transactions	Process of protecting data by ensuring the integrity of data during multi-user update sessions	Not supported	

## Data Service Connection Arguments for MySQL

To access data that is hosted on the Federation Server, a client must submit a connection string, which defines how to connect to the data. The data service connection arguments for MySQL include connection options and advanced options.

### Connection Options

Connection options are used to establish a connection to a data service. For each connection to the specified data service, one or more connection options should be specified in the create data service data definition language (DDL) statement. For more information on this DDL statement, see [Create Data Service](#).

The following connection options are supported by the Driver for MySQL:

- **BASEDIR=pathname;** — Specifies the base directory location where MySQL is installed. For example:  

```
basedir='drive:\ProgramFiles\mysql'
```
- **CATALOG=catalog-identifier;** — Specifies an arbitrary identifier for an SQL catalog, which groups logically related schemas. Any identifier is valid (for example, catalog=mysql).

**Requirement:** You must specify a catalog. For the MySQL database, this is a logical catalog name to use as an SQL catalog identifier.

**Interaction:** Connecting to multiple catalogs by specifying catalog=\* is not supported for the MySQL database.

- **DATABASE=database;** — Specifies the name of the MySQL database to which you want to connect. If the database name contains spaces or non-alphanumeric characters, you must enclose the database name in quotation marks. For example, database=sample.

**Requirement:** You must specify a database name.

**Alias:** DB

- **DRIVER=MYSQL;** — Identifies the data service to which you want to connect, which is a MySQL database.

**Requirement:** You must specify the driver.

- **HOST=address;** — Specifies the network IP (Internet Protocol) address of the computer that hosts the MySQL server.
- **PORT=number;** — Specifies the TCP/IP port that the MySQL server is listening to for connections. An example is port=5282.
- **SOCKET=path;** — Specifies the socket file to use for the connection.

## Advanced Options

Advanced driver options are additional options that are not required in order to connect to the data source. They are used to establish connections to catalogs, data source names (DSNs), and schemas. While advanced options can also be used when connecting to a data service, doing so will cause the specified options to apply to all data service connections.

The following advanced options are supported by the Driver for MySQL:

- **DBCREATE\_TABLE\_OPTS='DBMS-SQL-clauses';** — Specifies one or more database management system (DBMS) specific clauses that can be appended to the end of a CREATE TABLE statement. The Driver for MySQL passes the CREATE TABLE statement and its clauses to the DBMS, which executes the statement and creates the DBMS table.
- **DEFAULT\_TXN\_ISOLATION=REPEATABLE\_READ|READ\_COMMITTED|READ\_UNCOMMITTED|SERIALIZABLE;** — Specifies the default transaction isolation level for the connection. Transaction isolation levels specify the behavior for transactions when you are attempting to access the same tables as other simultaneous transactions. Therefore, transaction isolation levels control such transaction processing phenomena as dirty reads, nonrepeatable reads, and phantom reads.
  - **REPEATABLE\_READ** — Specifies that if a row is read once and that if, later in the same transaction, the row is read again, then that row is guaranteed not to have been updated or deleted by another concurrent transaction. Phantom reads, however, are not prevented.

**Alias:** RR

- **READ\_COMMITTED** — Specifies that a transaction cannot read a row that an earlier transaction changed but did not commit the change (thereby preventing a dirty read). However, the read might not be repeatable. This argument does not prevent reads in which a set of rows read once and then read again by the same transaction do not match. This is called a phantom read.

**Alias:** RC

- **READ\_UNCOMMITTED** — Specifies that a transaction can access uncommitted changes that have been made by other transactions (in either this application or in other applications). A transaction that uses the READ\_UNCOMMITTED isolation level cannot lock other transactions out of the row of data that it is reading. Therefore, transactions are not isolated from each other. If the READ\_UNCOMMITTED isolation level is used by a transaction that is working with a data service that supports other transaction isolation levels, the data service ignores the mechanism that it uses to implement those levels.

**Alias:** RU

- **SERIALIZABLE** — Specifies that all transactions interact only in ways that guarantee there is some serial ordering of the transactions that will return the same result. Serializable reads are repeatable and prevent both dirty reads and phantom reads. This selection provides the most data integrity protection but also costs the most in overhead because other transactions must wait until the current transaction finishes.

**Alias:** SER

**Default:** READ\_COMMITTED

- **DESCRIBE\_UTF8\_COLUMNS=CHAR|CHARUTF8|WCHAR;** — Specifies how the Driver for MySQL describes UTF-8 character columns to the application. By default, character columns that are created in the MySQL database are encoded in UTF-8 and therefore might contain Unicode data. Use this option to control how the Driver for MySQL describes these columns, based on the type of data that is stored in the columns.
  - **CHAR** — Specifies that the Driver for MySQL report UTF-8 character columns to the application as type CHAR with no encoding (CEI) value. This is the default. Use this value when the MySQL UTF-8 character columns contain only Latin 1 encoded data.
  - **CHARUTF8** — Specifies that the Driver for MySQL report UTF-8 character columns to the application as CHAR with an encoding (CEI) value of UTF-8. Use this value when the MySQL UTF-8 character columns contain non-Latin1 data. If the SAS® session encoding is not UTF-8, specifying the value CHARUTF8 causes the data to be transcoded into the SAS session encoding value, and the described column length might also be changed.
  - **WCHAR** — Specifies that the Driver for MySQL report UTF-8 character columns as WCHAR. This argument might cause errors if the application is moving the data to a data service such as a SAS data set that does not support WCHAR.

**Default:** CHAR

- **DRIVER\_TRACE='API|SQL|ALL';** — Requests tracing information, which logs transaction records to an external file that can be used for debugging purposes. The



Federation Server driver writes a record of each command that is sent to the DBMS to the trace log based on the specified tracing level, which determines the type of tracing information. The tracing levels are as follows:

- **ALL** — Activates all trace levels.
- **API** — Specifies that API method calls be sent to the trace log. This option is most useful if you are having a problem and need to send a trace log to DataFlux Technical Support for troubleshooting.
- **DRIVER** — Specifies that driver-specific information be sent to the trace log.
- **SQL** — Specifies that SQL statements that are sent to the DBMS be sent to the trace log. Tracing information is DBMS specific, but most Federation Server drivers log SQL statements such as SELECT and COMMIT.

**Default:** Tracing is not activated.

**Requirement:** If you activate tracing, you must also specify the location of the trace log with DRIVER\_TRACEFILE=. (Optional) You can control trace log formatting with DRIVER\_TRACEOPTIONS=.

**Interaction:** You can specify one trace level, or you can concatenate more than one by including the | (OR) symbol. For example: driver\_trace='api|sql' specifies to generate tracing information for API calls and SQL statements.

- **DRIVER\_TRACEFILE='filename';** — Specifies where to write trace information. The trace log is a text file. Include the filename and extension in single or double quotation marks. For example:

```
driver_tracefile='\mytrace.log'
```

**Default:** There is no default trace log location.

**Requirement:** You must specify a location for a trace log if you activate tracing with DRIVER\_TRACE=.

**Interaction:** (Optional) You can control trace log formatting with DRIVER\_TRACEOPTIONS=.

- **DRIVER\_TRACEOPTIONS=APPEND|THREADSTAMP|TIMESTAMP;** — Specifies options in order to control formatting and other properties for the trace log:
  - **APPEND** — Adds trace information to the end of an existing trace log. The contents of the file are not overwritten.
  - **THREADSTAMP** — Prepends each line of the trace log with a thread identification.
  - **TIMESTAMP** — Prepends each line of the trace log with a time stamp.

**Default:** The trace log is overwritten with no thread identification or time stamp.

- **PASSWORD=password;** — Specifies the password that corresponds to the user ID in the MySQL database. If the password contains spaces or non-alphanumeric characters, you must enclose the password in quotation marks.
  - **Alias:** PWD=

- **STORE\_RESULTS=YES|NO;** — Specifies whether to transfer a result set from the last query.
- **USER=id;** — Specifies the user ID for logging on to the MySQL server. If this argument is not specified, the current user is assumed. If the user ID contains spaces or non-alphanumeric characters, you must enclose the user ID in quotation marks.
- **Alias:** UID=

## Data Types for MySQL

When an application submits Federation Server language statements, only the Federation Server data types are supported for column definition. Native MySQL data types that are not supported are mapped to similar Federation Server data types for data retrieval. For information about the Federation Server data types, see [Data Service Data Types](#).

When an application submits MySQL SQL statements, native MySQL data types are supported. Submitting MySQL SQL statements is available only to applications that consume the JDBC, ODBC, or OLE DB client-side drivers. For information about the native MySQL SQL data types, see your MySQL documentation.

The following table lists the data types that are supported for MySQL:

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
BIGINT	BIGINT	Specifies a large signed, exact whole number with a precision of 19.	BIGINT
*	BLOB	Specifies a varying-length binary large object string up to 64 kilobytes.	*
CHAR( <i>n</i> )	CHAR( <i>n</i> )	Specifies a fixed-length character string from 1 to 255 characters; the default is 1.	CHAR( <i>n</i> )
DATE	DATE	Specifies a date value in the range January 1, 1000 to December 31, 9999, in the format <i>YYYY-MM-DD</i> .	DATE
*	DATETIME	Specifies a date and time value in the range 00:00:00 on January 1, 1000 to 23:59:59 on December 31, 9999, in the format <i>YYYY-MM-DD HH:MM:SS</i> .	*
*	DECIMAL( <i>p,s</i> )	Specifies a signed, fixed-point decimal number.	*
DOUBLE	DOUBLE	Specifies a signed, double precision, floating-point number with a precision of 53.	DOUBLE

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
*	ENUM( <i>values</i> )	Specifies a character value that can be chosen from the list of allowed values. You can specify up to 65,535 ENUM values.	*
FLOAT( <i>p</i> )	FLOAT( <i>p</i> )	Specifies a signed, single precision or double precision, floating-point number with a precision range of 1 to 24 (single precision) or 25 to 53 (double precision); the default is 53.	FLOAT( <i>p</i> )
INTEGER	INT	Specifies a regular signed, exact whole number with a precision of 10.	INTEGER
*	LONGBLOB	Specifies a varying-length binary data up to 4 gigabytes.	*
*	LONGTEXT	Specifies a varying-length character string up to 4 gigabytes.	*
*	MEDIUMBLOB	Specifies a varying-length binary data up to 16 megabytes.	*
*	MEDIUMINT		*
*	MEDIUMTEXT	Specifies a varying-length character string up to 16 megabytes.	*
*	SET( <i>values</i> )	Specifies a character value that must be chosen from the list of allowed values. You can specify up to 64 SET values.	*
SMALLINT	SMALLINT	Specifies a small signed, exact whole number with a precision of 5.	SMALLINT
*	TEXT	Specifies a varying-length text data up to 64 kilobytes.	*
TIME	TIME	Specifies a time value in the range - 838 hours, 59 minutes, 59 seconds to 838 hours, 59 minutes, 59 seconds in the format <i>HH:MM:SS</i> .	TIME
TIMESTAMP	TIMESTAMP	Specifies a date and time value in the range from 00:00:00 on January 1, 1970 to 2037 in the format <i>YYYY-MM-DD HH:MM:SS</i> .	TIMESTAMP
*	TINYBLOB	Specifies a varying-length binary large object string up to 256 bytes.	*
TINYINT	TINYINT	Specifies a very small signed, exact whole number with a precision of 3.	TINYINT
*	TINYTEXT	Specifies a varying-length text data	*

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
		up to 256 bytes.	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Specifies a varying-length character string from 1 to 255 characters.	VARCHAR( <i>n</i> )

\* For the Federation Server languages, the data type cannot be defined, and when data is retrieved, the Federation Server maps the native MySQL data type to a similar Federation Server data type.

## Naming Conventions for MySQL

MySQL database identifiers that can be named include databases, tables, and columns. The MySQL documentation contains extensive information about naming conventions. Here are some of the naming conventions that you must use:

- All identifier names must be from 1 to 64 characters long, except for aliases, which can be 255 characters long.
- Database names must be unique. For each user within a database, names of database objects must be unique across all users (for example, if a database contains a department table created by user A, no other user can create a department table in the same database).



**Note:** MySQL does not support schemas. Consequently, tables are automatically visible to all users who have appropriate privileges. Column names and index names must be unique within a table.

- Database names can use any character that is allowed in a directory name except for periods and backward and forward slashes.
- Table names can use any character that is allowed in a filename except for periods and forward slashes.
- Column and alias names allow all characters.
- A name cannot be a MySQL reserved word unless the name is enclosed in quotation marks. See the MySQL documentation for more information about reserved words.
- Embedded spaces and other special characters are not permitted unless the name is enclosed in quotation marks.
- Embedded quotation marks are not permitted.
- Case sensitivity is set when a server is installed. By default, the names of database objects are case sensitive on UNIX® and not case sensitive on Windows®. For example, the names CUSTOMER and customer are different on a case-sensitive server.

# TSSQL Statement Table Options

Federation Server TSSQL (TSSQL) statement table options specify actions that apply only to the table with which they appear. They enable you to perform operations such as specifying a password for the driver file and renaming columns.

You can specify table options on a TSSQL statement on which you specify a table name, such as the CREATE TABLE, ALTER TABLE, or SELECT statement.

For more information on how to use these options, see "Using TSSQL Statement Table Options" in the *DataFlux Federation Server TSSQL Reference Guide*.

## Syntax

Specify a TSSQL statement table option immediately after the table name, within curly braces { } and including the keyword OPTION. To specify several table options, separate them with spaces or commas.

```
{OPTION option-1=value [... option-n=value] }
```



**Important:** While specifying the syntax for table options, you cannot have a space between the left curly brace { and the keyword OPTION. A space will result in a syntax error.

These examples show table options in TSSQL statements:

```
create table salary {option encrypt=yes read=green};  
select * from salary {option read=green};
```

The [DBCREATE\\_TABLE\\_OPTS= Table Option](#) is supported by the Driver for MySQL.

## DBCREATE\_TABLE\_OPTS= Table Option

Specifies DBMS-specific syntax to be added to the CREATE TABLE statement.

### Syntax

```
DBCREATE_TABLE_OPTS= 'DBMS-SQL-clauses'
```

### Syntax Description

- **DBMS-SQL-clauses** — Specifies one or more DBMS-specific clauses that can be appended to the end of an SQL CREATE TABLE statement.

### Details

This option enables you to add DBMS-specific clauses to the end of the CREATE TABLE statement. The interface passes the CREATE TABLE statement and its clauses to the DBMS, which executes the statement and creates the DBMS table.

## Examples

In the following example, the table TEMP is created with the value of the DBCREATE\_TABLE\_OPTS= option appended to the CREATE TABLE statement:

```
create table temp {option dbcreate_table_opts='partitioning key (x) using  
hashing'} (x double);
```

The following CREATE TABLE statement is passed to the DBMS in order to create the table:

```
create table temp (x double) partitioning key (x) using hashing;
```

# ODBC Reference

This section provides functionality details and guidelines for the open database connectivity (ODBC) databases that are supported by the DataFlux® Driver for ODBC (Driver for ODBC).

ODBC standards provide a common interface to a variety of databases, including dBASE, Microsoft® Access®, Oracle®, Paradox, and Microsoft SQL Server databases. Specifically, ODBC standards define APIs that enable an application to access a database if both the application and the database conform to the specification. ODBC also provides a mechanism to enable dynamic selection of a database that an application is accessing, so that users have the flexibility of selecting databases other than those that are specified by the application developer.

The following topics provide additional information about the Driver for ODBC:

- [Understanding the Driver for ODBC](#)
- [Database Functionality Support for ODBC](#)
- [Data Service Connection Arguments for ODBC](#)
- [Data Types for ODBC](#)
- [Naming Conventions for ODBC](#)

## Understanding the Driver for ODBC

The Driver for ODBC enables the Federation Server to read and update legacy ODBC database tables. In addition, the driver creates tables that can be accessed by both the Federation Server and an ODBC database.

The Driver for ODBC supports most of the TSSQL functionality. The driver also supports an application's ability to submit native database-specific SQL statements.

The Driver for ODBC is a remote driver, which means that it connects to a server process in order to access data. The process might be running on the same machine as the Federation Server, or it might be running on another machine in the network.

## Database Functionality Support for ODBC

The following table lists the database functionality support level for an ODBC database that is provided by the Driver for ODBC. For general information about database functionality, see [Database Functionality Supported by Data Services](#).

Functionality	Description	Support	Comment
bookmark	Stored location used for quick retrieval and scrollable cursors	Supported	
bulk operations	Ability to add, delete, or update a large quantity of data with one operation	Supports additions, deletions, and updates	

Functionality	Description	Support	Comment
cursor support	Functionality that keeps track of the position in a result set and enables multiple operations on the result set	dependent on the database management system (DBMS); not all cursor types are supported	Cursor functionality is supported by the driver or by specifying that the Federation Server provides additional cursor support.
index	Component of a database that contains information about the physical location of data stored in a file		
integrity constraint	Functionality that ensures correctness, completeness, and compliance of data	Supported	
multi-row fetch	Enables an application to request more than one row at a time	Supported	
stored procedures	SQL programs that perform computations or other tasks consistently	Result set stored procedures with parameters	
threaded I/O processing	Efficient method of processing a query with multiple threads	Supported	
transactions	Process of protecting data by ensuring the integrity of data during multi-user update sessions	Supported	

## Data Service Connection Arguments for ODBC

To access data that is hosted on the Federation Server, a client must submit a connection string, which defines how to connect to the data. The data service connection arguments for an ODBC-compliant database include connection options and advanced options.

### Connection Options

Connection options are used to establish a connection to a data service. For each connection to the specified data service, one or more connection options should be specified in the create data service data definition language (DDL) statement. For more information on this DDL statement, see [Create Data Service](#).

The following connection options are supported for an ODBC-compliant database:

- **CATALOG=catalog-identifier;** — Specifies an arbitrary identifier for an SQL catalog, which groups logically related schemas. For databases that do not support native catalogs, any identifier is valid (for example, catalog=myodbc). For the SQL Server, which is a multi-catalog database, CATALOG= is not required. The connection defaults to CATALOG=\* unless you specify a logical name for the catalog and map it to the



native catalog name in the SQL Server. For example, to map the logical catalog mycat to the native catalog named newusers, specify the following code:

```
catalog=(mycat=newusers);
```

**Requirement:** You must specify a catalog name for all databases except the SQL Server, which can omit the CATALOG= option, or else specify a catalog name map. Catalog name maps can be used with TSSQL only. They cannot be used to access native DBMS SQL.

- **CONOPTS=(valid-ODBC-compliant-database-connection-string);** — Specifies, within parentheses, an ODBC-compliant database connection string, including any valid Driver for ODBC connection options that are not provided by the Federation Server connection options.

For connections that do not use a DSN, include the ODBC-specific DRIVER= keyword; that is, if you do not include a DSN= connection option to specify an ODBC DSN. For example, specify DRIVER={SQL Server} followed by appropriate connection options that are valid in SQL Server connections.

If you include a DSN= or FILEDSN= specification with the CONOPTS= option, do not use the ODBC\_DSN= connection option. However, you can specify the ODBC database-specific connection options by using CONOPTS= and then you can specify an ODBC DSN that contains other connection information by using the ODBC\_DSN= connection option.

- **DRIVER=ODBC;** — Calls the Driver for ODBC. This specifies that the data service to which you want to connect must be an ODBC-compliant database.

**Requirement:** You must specify the driver.

- **ODBC\_DSN=ODBC DSN name;** — Specifies a valid ODBC-compliant database DSN that contains connection information for connecting to the ODBC-compliant database. You can use the CONOPTS= option in addition to ODBC\_DSN= option to specify database-specific connection options not provided by the Federation Server. Do not specify the ODBC DSN in both CONOPTS= and ODBC\_DSN=.

## Advanced Options

Advanced driver options are additional options that are not required in order to connect to the data source. They are used to establish connections to catalogs, data source names (DSNs), and schemas. While advanced options can also be used when connecting to a data service, doing so will cause the specified options to apply to all data service connections.

The following advanced options are supported for an ODBC-compliant database:

- **DEFAULT\_CURSOR\_TYPE=FORWARD\_ONLY|KEYSET\_DRIVEN|DYNAMIC|STATIC;** — Specifies a valid default cursor type for new statements. The valid options are:
  - **FORWARD\_ONLY** — Specifies a non-scrollable cursor that moves only forward through the result set. Forward-only cursors are dynamic in that all changes are detected as the current row is processed. If an application does not require scrolling, the forward-only cursor retrieves data quickly, with the least amount of overhead processing.

- **KEYSET\_DRIVEN** — Specifies a scrollable cursor that detects changes that are made to the values of rows in the result set but that does not always detect changes to deletion of rows and changes to the order of rows in the result set. A keyset-driven cursor is based on row keys, which are used to determine the order and set of rows that are included in the result set. As the cursor scrolls the result set, it uses the keys to retrieve the most recent values in the table.

It is sometimes helpful to have a cursor that can detect changes in the rows of a result set. A keyset-driven cursor uses a row identifier rather than caching the entire row into memory. It therefore uses much less disk space than other row caching mechanisms. Deleted rows can be detected when a SELECT statement that references the bookmark/row ID/key column values fails to return a row.

- **DYNAMIC** — Specifies a scrollable cursor that detects changes that are made to the rows in the result set. All INSERT, UPDATE, and DELETE statements that are made by all users are visible through the cursor. The dynamic cursor is good for an application that must detect all concurrent updates that are made by other users.
- **STATIC** — Specifies a scrollable cursor that displays the result set as it existed when the cursor was first opened. The static cursor provides forward and backward scrolling. If the application does not need to detect changes but requires scrolling, the static cursor is a good choice.

**Default:** There is no default value.



**Note:** The application can still override this value, but if the application does not explicitly set a cursor type, this value will be in effect.

- **DRIVER\_TRACE='API|SQL|ALL';** — Requests tracing information, which logs transaction records to an external file that can be used for debugging purposes. The Federation Server driver writes a record of each command that is sent to the DBMS to the trace log based on the specified tracing level, which determines the type of tracing information. The tracing levels are as follows:
  - **ALL** — Activates all trace levels.
  - **API** — Specifies that API method calls be sent to the trace log. This option is most useful if you are having a problem and need to send a trace log to DataFlux Technical Support for troubleshooting.
  - **DRIVER** — Specifies that driver-specific information be sent to the trace log.
  - **SQL** — Specifies that SQL statements that are sent to the DBMS be sent to the trace log. Tracing information is DBMS specific, but most Federation Server drivers log SQL statements such as SELECT and COMMIT.

**Default:** Tracing is not activated.

**Requirement:** If you activate tracing, you must also specify the location of the trace log with DRIVER\_TRACEFILE=. (Optional) You can control trace log formatting with DRIVER\_TRACEOPTIONS=.

**Interaction:** You can specify one trace level, or you can concatenate more than one by including the | (OR) symbol. For example: driver\_trace='api|sql' specifies to generate tracing information for API calls and SQL statements.

- **DRIVER\_TRACEOPTIONS=APPEND|THREADSTAMP|TIMESTAMP;** — Specifies options in order to control formatting and other properties for the trace log:
  - **APPEND** — Adds trace information to the end of an existing trace log. The contents of the file are not overwritten.
  - **THREADSTAMP** — Prepends each line of the trace log with a thread identification.
  - **TIMESTAMP** — Prepends each line of the trace log with a time stamp.

**Default:** The trace log is overwritten with no thread identification or time stamp.
- **PASSWORD=password;** — Specifies the password that corresponds to the user ID in the database.
  - **Alias:** PWD=
- **USER=id;** — Specifies the user ID for logging on to the ODBC-compliant database, such as Microsoft SQL Server, with a user ID that is different than the default ID.
  - **Alias:** UID=

Examples of valid connection strings are as follows:

This connection string specifies an ODBC DSN:

```
driver=odbc; uid=scott; pw=roger; odbc_dsn=myOracleDSN;
catalog=odbc_oracle;
```

This connection string specifies catalog name maps to access multiple catalogs on Microsoft SQL Server:

```
driver=odbc; uid=jfox; pw=mypw; odbc_dsn=mySQLdsn;
catalog=(cat1=mycat; cat2=testcat; cat3=users;
```

## Data Types for ODBC

When an application submits Federation Server language statements, only the Federation Server data types are supported for column definitions. Native database data types that are not supported data types are mapped to similar Federation Server data types for data retrieval. For information about the Federation Server data types, see [Data Service Data Types](#).

When an application submits native SQL statements, native database data types are supported. Submitting native SQL statements is available only to applications that consume the JDBC, ODBC, or OLE DB client-side drivers. For information about the native database SQL data types, see your database documentation.

The following table lists the data types that are supported for ODBC databases:

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
BIGINT	BIGINT	Specifies a large signed, exact whole number with a precision of 19.	BIGINT
CHAR( <i>n</i> )	CHAR( <i>n</i> )	Specifies a fixed-length character string from 1 to 200 characters; the default is 1.	CHAR( <i>n</i> )
DATE	DATE	Specifies a date value in the range from 1582 AD to 9999 AD in the format <i>mm-dd-yyyy</i> .	DATE
*	DECIMAL( <i>p,s</i> )	Specifies a signed, fixed-point decimal number. Precision range is 1 to 31; the default is 5. Scale range is 1 to 31; the default is 0.	*
DOUBLE	DOUBLE	Specifies a signed, double precision, floating-point number with a precision of 53.	DOUBLE
INTEGER	INT	Specifies a regular signed, exact whole number with a precision of 10.	INTEGER
NCHAR( <i>n</i> )	NCHAR( <i>n</i> )	Specifies a fixed-length character string.	NCHAR( <i>n</i> )
*	NUMERIC( <i>p,s</i> )	Specifies a NUMERIC( <i>p,s</i> ) Signed, fixed-point decimal number. Precision range is 1 to 38. Scale range is -84 to 127; the default is 0.	*
NVARCHAR( <i>n</i> )	NVARCHAR( <i>n</i> )	Specifies a varying-length character string.	NVARCHAR( <i>n</i> )
SMALLINT	SMALLINT	Specifies a small signed, exact whole number with a precision of 5.	SMALLINT
TIME( <i>p</i> )	TIME( <i>p</i> )	Specifies a time value from 00:00:00 to 23:59:59.	TIME( <i>p</i> )
TIMESTAMP( <i>p</i> )	TIMESTAMP( <i>p</i> )	Specifies a date, time, and optional microsecond value in the range from 1582 AD to 9999 AD with 24-hour day (00:00:00 to 23:59:59).	TIMESTAMP( <i>p</i> )
TINYINT	TINYINT	Specifies a very small signed, exact whole number with a precision of 3.	TINYINT

<b>Federation Server Language Keyword</b>	<b>Data Service Type Name</b>	<b>Data Service Type Definition</b>	<b>Federation Server Type Returned</b>
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Specifies a varying-length character string from 1 to 255 characters.	VARCHAR( <i>n</i> )

\* For the Federation Server languages, the data type cannot be defined, and when data is retrieved, the Federation Server maps the native database data type to a similar Federation Server data type.

## Naming Conventions for ODBC

Because ODBC is not a database but rather is an API, table and column names are determined at run time. The Driver for ODBC supports table and column names that are up to 32 characters long. If the DBMS column names are longer than 32 characters, they are truncated to 32 characters. If truncating a column's name results in identical names, then unique names are generated by replacing the last character with a number. DBMS table names must be 32 or fewer characters because DataFlux does not truncate a longer name. If you already have a table name that is longer than 32 characters, it is recommended that you create a table view.

# Oracle Reference

This section provides functionality details and guidelines for the Oracle® tables that are supported by the DataFlux® Driver for Oracle (Driver for Oracle).

The following topics provide additional information about the Driver for Oracle:

- [Understanding the Driver for Oracle](#)
- [Database Functionality Support for Oracle](#)
- [Data Service Connection Arguments for Oracle](#)
- [Data Types for Oracle](#)
- [Load Data by Using Bulk Operations for Oracle](#)
- [Naming Conventions for Oracle](#)
- [TSSQL Statement Table Options](#)

## Understanding the Driver for Oracle

The Driver for Oracle enables the Federation Server to read and update legacy Oracle tables. In addition, the driver creates Oracle tables that can be accessed by both the Federation Server and Oracle.

The Driver for Oracle supports most of the TSSQL functionality. The driver also supports the application's ability to submit native Oracle SQL statements.

The Driver for Oracle is a remote driver, which means that it connects to a server process in order to access data. The process might be running on the same machine as the Federation Server, or it might be running on another machine in the network.

## Database Functionality Support for Oracle

The following table lists the database functionality support level for Oracle that is provided by the Driver for Oracle. For general information about database functionality, see [Database Functionality Supported by Data Services](#).

Functionality	Description	Support	Comment
bookmark	Stored location used for quick retrieval and scrollable cursors	Supported	
bulk operations	Ability to add, delete, or update a large quantity of data with one operation	Supports additions, deletions, and updates	For more information, see <a href="#">Load Data by Using Bulk Operations for Oracle</a> .
cursor support	Functionality that keeps track of the position in a result set and enables	Static cursor	Cursor functionality is supported by the driver or by specifying that the

Functionality	Description	Support	Comment
	multiple operations on the result set		Federation Server provides additional cursor support.
index	Component of a database that contains information about the physical location of data stored in a file		
integrity constraint	Functionality that ensures correctness, completeness, and compliance of data	Supported	
multi-row fetch	Enables an application to request more than one row at a time	Supported	
stored routines	SQL programs that perform computations or other tasks consistently	Stored procedures with parameters	
threaded I/O processing	Efficient method of processing a query with multiple threads	Not supported	
transactions	Process of protecting data by ensuring the integrity of data during multi-user update sessions	Supported	

## Data Service Connection Arguments for Oracle

To access data that is hosted on the Federation Server, a client must submit a connection string, which defines how to connect to the data. The data service connection arguments for Oracle Server include connection options and advanced options.

### Connection Options

Connection options are used to establish a connection to a data service. For each connection to the specified data service, one or more connection options should be specified in the create data service data definition language (DDL) statement. For more information on this DDL statement, see [Create Data Service](#).

The following connection options are supported for Oracle Server:

- **CATALOG=catalog-identifier;** — Specifies an arbitrary identifier for an SQL catalog, which groups logically related schemas. Any identifier is valid (for example, catalog=oracle\_test).
- Requirement:** You must specify a catalog. For the Oracle database, this is a logical catalog name to use as an SQL catalog identifier.
- **DRIVER=ORACLE;** — Identifies the data service to which you want to connect, which is an Oracle database.

**Requirement:** You must specify the driver.

- **ORNUMERIC=NO|YES** — Specifies how numbers read from or inserted into the Oracle NUMBER column will be treated. This option can be specified as both a connection option and a table option. When specified as both connection and table option, the table option value overrides the connection option.
  - **NO** — Indicates that the numbers will be treated as TKTS\_DOUBLE values. They may not have precision beyond 14 digits.
  - **YES** — Indicates that non-integer values with explicit precision will be treated as TKTS\_NUMERIC values.

**Default:** YES

**Examples:**

```
libname basedir '.';

proc tssql
conn="(DRIVER=SQL;CONOPTS=((DRIVER=ORACLE;UID=scott;PWD=tiger;
PATH=tktsora;ORNUMERIC=NO;CATALOG=ORACAT;);(DRIVER=BASE;SCHEMA=(NAME=one;P
RIMARY PATH='.';CATALOG=BASECAT) ));";
drop table basecat.temp;
create table basecat.temp as select * from oracat.TEMP; /* pick
ORNUMERIC=NO from connection option */
quit;

proc tssql
conn="(DRIVER=SQL;CONOPTS=((DRIVER=ORACLE;UID=scott;PWD=tiger;
PATH=tktsora;ORNUMERIC=NO;CATALOG=ORACAT;);(DRIVER=BASE;SCHEMA=(NAME=one;P
RIMARY PATH='.';CATALOG=BASECAT) ));";
drop table basecat.temp;
create table basecat.temp as select * from oracat.TEMP{option
ORNUMERIC=YES}; /*override connection option NO with Table option YES */
quit;

proc tssql
conn="(DRIVER=SQL;CONOPTS=((DRIVER=ORACLE;UID=scott;PWD=tiger;
PATH=tktsora;CATALOG=ORACAT;);(DRIVER=BASE;SCHEMA=(NAME=one;PRIMARYPATH='.
';CATALOG=BASECAT) ));";
drop table basecat.temp;
create table basecat.temp as select * from oracat.TEMP; /*default to
ORNUMERIC=YES*/
quit;

proc tssql
conn="(DRIVER=SQL;CONOPTS=((DRIVER=ORACLE;UID=scott;PWD=tiger;
PATH=tktsora;CATALOG=ORACAT;);(DRIVER=BASE;SCHEMA=(NAME=one;PRIMARYPATH='.
';CATALOG=BASECAT) ));";
drop table basecat.temp;
create table basecat.temp as select * from oracat.TEMP{option
ORNUMERIC=NO}; /*override default value YES with NO*/
quit;
```

- **PATH=database-specification;** — Specifies the Oracle connect identifier. A connect identifier can be a net service name, a database service name, or a net service alias.

## Advanced Options

Advanced driver options are additional options that are not required in order to connect to the data source. They are used to establish connections to catalogs, data source names



(DSNs), and schemas. While advanced options can also be used when connecting to a data service, doing so will cause the specified options to apply to all data service connections.

The following advanced options are supported for Oracle Server:

- **DRIVER\_TRACE='API|SQL|ALL';** — Requests tracing information, which logs transaction records to an external file that can be used for debugging purposes. The Federation Server driver writes a record of each command that is sent to the database to the trace log based on the specified tracing level, which determines the type of tracing information. The tracing levels are as follows:
  - **ALL** — Activates all trace levels.
  - **API** — Specifies that API method calls be sent to the trace log. This option is most useful if you are having a problem and need to send a trace log to DataFlux Technical Support for troubleshooting.
  - **DRIVER** — Specifies that driver-specific information be sent to the trace log.
  - **SQL** — Specifies that SQL statements that are sent to the database management system (DBMS) be sent to the trace log. Tracing information is DBMS specific, but most Federation Server drivers log SQL statements such as SELECT and COMMIT.

**Default:** Tracing is not activated.

**Requirement:** If you activate tracing, you must also specify the location of the trace log with DRIVER\_TRACEFILE=. (Optional) You can control trace log formatting with DRIVER\_TRACEOPTIONS=.

**Interaction:** You can specify one trace level, or you can concatenate more than one by including the | (OR) symbol. For example: driver\_trace='api|sql' specifies to generate tracing information for API calls and SQL statements.

- **DRIVER\_TRACEFILE='filename';** — Specifies where to write trace information. The trace log is a text file. Include the filename and extension in single or double quotation marks. For example:

```
driver_tracefile='\\mytrace.log'
```

**Default:** There is no default trace log location.

**Requirement:** You must specify a location for a trace log if you activate tracing with DRIVER\_TRACE=.

**Interaction:** (Optional) You can control trace log formatting with DRIVER\_TRACEOPTIONS=.

- **DRIVER\_TRACEOPTIONS=APPEND|THREADSTAMP|TIMESTAMP;** — Specifies options in order to control formatting and other properties for the trace log:
  - **APPEND** — Adds trace information to the end of an existing trace log. The contents of the file are not overwritten.
  - **THREADSTAMP** — Prepends each line of the trace log with a thread identification.
  - **TIMESTAMP** — Prepends each line of the trace log with a time stamp.

**Default:** The trace log is overwritten with no thread identification or time stamp.

- **ORA\_ENCODING=UNICODE;** — Specifies that the Oracle data be returned in Unicode to the Federation Server. By default Oracle data is returned in WLATIN1, independent of the NLS\_LANG environment variable setting. This option must be specified if the Federation Server or SAS client is encoded for multibyte encoding because data is lost during the transcoding to single-byte encoding. For example, a SAS client in a Chinese environment cannot read data in an Oracle UTF-8 database unless this option is set, which causes the Unicode data to be successfully transcribed to Chinese by the Federation Server.
- **PWD=password;** — Specifies an optional Oracle database password that is associated with the Oracle user ID. PWD= is always used with UID= and the associated password is case-sensitive. If you omit PWD=, the password for the default Oracle user ID **OPS\$sysid** is used, if it is active.
- **UID=userid;** — Specifies an optional Oracle user ID. If the user ID contains blanks or national characters, enclose it in quotation marks. If you omit an Oracle user ID and password, the default Oracle user ID OPS\$sysid is used, if it is enabled. UID= must be used with PWD=.
- **USE\_CACHED\_CATALOG=YES|NO;** — Specifies whether to use the cached catalog rather than compiling a new catalog on every run. Setting this option to YES can improve the performance of the TKTSTForeignKeys API.

Before you can use this option, you must complete the following steps:

1. Create a materialized view. For a sample script showing how to create a materialized view, see [Appendix C: Code Example](#).
2. Use the ALTER DSN statement to add the USE\_CACHED\_CATALOG connection option. For more information on the ALTER DSN statement, see [ALTER DSN Statement](#).

**Default:** YES

## Data Types for Oracle

When an application submits TSSQL statements, only the Federation Server data types are supported for column definition. Native Oracle data types that are not supported by the Federation Server are mapped to similar Federation Server data types for data retrieval. For information about the Federation Server data types, see [Data Service Data Types](#).

When an application submits Oracle SQL statements, native Oracle data types are supported. Submitting Oracle SQL statements is available only to applications that consume the JDBC, ODBC, or OLE DB client-side drivers. For information about the native Oracle SQL data types, see your Oracle documentation.

The following table lists the data types that are supported for Oracle:

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
BIGINT	BIGINT	Specifies a large signed, exact whole number with a precision of 19.	BIGINT
CHAR( <i>n</i> )	CHAR( <i>n</i> )	Specifies a fixed-length character string from 1 to 255 (Oracle 7) 2,000 (Oracle 8) characters; the default is 1. Note that the Oracle7 Server CHAR data type is not equivalent to the Oracle Version 6 CHAR data type. The Oracle7 Server CHAR data type is new with the Oracle7 Server and uses blank-padded comparison semantics.	CHAR
DATE	DATE	Specifies a date value in the range 0001-01-01 to 9999-12-31.	DATE
*	DECIMAL( <i>p,s</i> )	Specifies a signed, fixed-point decimal number. Precision range is 1 to 31; the default is 5. Scale range is 1 to 31; the default is 0. Precision can be greater than what is specified; scale must be less than or	*

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
		equal to precision.	
DOUBLE	DOUBLE	Specifies a signed, double precision, floating-point number with a precision of 53.	DOUBLE
FLOAT( <i>p</i> )	FLOAT( <i>p</i> )	Signed, single precision or double precision floating-point number with a precision range of 1 to 24 (single precision) or 25 to 53 (double precision).	FLOAT
INTEGER	INTEGER	Specifies a regular signed, exact whole number with a precision of 10.	INTEGER
NCHAR( <i>n</i> )	NCHAR( <i>n</i> )		NCHAR
NUMERIC	NUMBER	Specifies a signed, double precision, floating-point number with a precision of 38.	NUMERIC
NVARCHAR( <i>n</i> )	NVARCHAR( <i>n</i> )		NVARCHAR
REAL	REAL	Specifies a signed, single precision floating-point number with a precision of 24.	REAL
SMALLINT	SMALLINT	Specifies a small signed, exact whole number with a precision of 5.	SMALLINT
TIME( <i>p</i> )	TIME( <i>p</i> )	Specifies a time value in the range 00:00:00.000000 to 0	TIME

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
		23:59:61.999999.	
TIMESTAMP( <i>p</i> )	TIMESTAMP( <i>p</i> )	Specifies a date and time value in range 0001-01-01:00:00:00.000 000 to 9999-12-31:23:59:61.999 999.	TIMESTAMP
TINYINT	TINYINT	Specifies a very small signed, exact whole number with a precision of 3.	TINYINT
VARBINARY( <i>n</i> )	VARBINARY( <i>n</i> )	Specifies a varying-length binary string.	VARBINARY
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Specifies a varying-length character string from 1 to 2,000 characters for Oracle 7 and 1 to 4,000 characters for Oracle 8.	VARCHAR

\* For the Federation Server languages, the data type cannot be defined, and when data is retrieved, the Federation Server maps the native Oracle data type to a similar Federation Server data type.

## Load Data by Using Bulk Operations for Oracle

To use the Oracle bulk-load API, which improves performance for large amounts of data, specify the TSSQL table option BULKLOAD=YES.

## Naming Conventions for Oracle

Oracle objects that can be named include tables, views, columns, and indexes. For the Oracle7 Server, objects also include database triggers, procedures, and stored functions.

Use the following Oracle naming conventions:

- A name must start with a letter. However, if the name appears within double quotation marks, it can start with any character.

- A name must be from 1 to 30 characters long, except for database names, which are limited to 8 characters, and link names, which are limited to 128 characters.
- A name can contain the letters A through Z, the digits 0 through 9, the underscore (\_), \$, and #. If the name appears within double quotation marks, it can contain any characters, except double quotation marks.
- A name is not case sensitive. For example, CUSTOMER is the same as customer. However, if the name of the object appears within double quotation marks when it is used, then it is case sensitive.
- A name cannot be an Oracle reserved word.
- A name cannot be the same name as another Oracle object in the same schema.

## TSSQL Statement Table Options

Federation Server TSSQL (TSSQL) statement table options specify actions that apply only to the table with which they appear. They enable you to perform operations such as specifying a password for the driver file and renaming columns.

You can specify table options on a TSSQL statement on which you specify a table name, such as the CREATE TABLE, ALTER TABLE, or SELECT statement.

For more information on how to use these options, see "Using TSSQL Statement Table Options" in the *DataFlux Federation Server TSSQL Reference Guide*.

## Syntax

Specify a TSSQL statement table option immediately after the table name, within curly braces { } and including the keyword OPTION. To specify several table options, separate them with spaces or commas.

```
{OPTION option-1=value [... option-n=value] }
```



**Important:** While specifying the syntax for table options, you cannot have a space between the left curly brace { and the keyword OPTION. A space will result in a syntax error.

These examples show table options in TSSQL statements:

```
create table salary {option encrypt=yes read=green};
select * from salary {option read=green};
```

The following statement table options are supported by the Driver for Oracle:

- [BL\\_BULKOPTS= Table Option](#)
- [BL\\_DEFAULT\\_DIR= Table Option](#)
- [BL\\_ERRORS= Table Option](#)
- [BL\\_LOAD= Table Option](#)
- [BL\\_LOGFILE= Table Option](#)
- [BL\\_PARALLEL= Table Option](#)

- [BL\\_RECOVERABLE= Table Option](#)
- [BL\\_SKIP= Table Option](#)
- [BL\\_SKIP\\_INDEX\\_MAINTENANCE= Table Option](#)
- [BL\\_SKIP\\_UNUSABLE\\_INDEXES= Table Option](#)
- [BULKLOAD= Table Option](#)

## BL\_BULKOPTS= Table Option

Container option for the bulkload options. The category is Bulk Loading. This option must follow BULKLOAD=YES. For more information, see [BULKLOAD= Table Option](#).

### Syntax

```
BL_BULKOPTS= (option [, ...option ])
```

### Syntax Description

- *option* — Specifies one or more bulkload table options.

### Details

The BL\_BULKOPTS= option is a container option that is required in order to specify the other bulkload options. To specify BL\_BULKOPTS= option, you must first specify BULKLOAD=YES. For more information, see [BULKLOAD= Table Option](#).

## BL\_DEFAULT\_DIR= Table Option

Specifies the directory where the bulk load log file is created.

### Syntax

```
BL_DEFAULT_DIR='dir path'
```

### Syntax Description

- ***dir path*** — Indicates the directory in which the log file is created for the load. The default is the current directory.

## BL\_ERRORS= Table Option

Specifies that after the indicated number of errors is received, that the load should stop.

### Syntax

```
BL_ERRORS=n
```

### Syntax Description

- ***n*** — Indicates the number of errors that should be received before the load stops. The default is 1000000.

## BL\_LOAD= Table Option

Specifies that after the indicated number of rows is loaded, that the load should stop.

### Syntax

```
BL_LOAD=n
```

### Syntax Description

- ***n*** — Indicates the number of rows that should be loaded; the first *n* rows will be loaded. The default is to load all rows.

## BL\_LOGFILE= Table Option

Indicates the file name for the bulk load log file.

### Syntax

```
BL_LOGFILE='log file name'
```

### Syntax Description

- **log file name** — Specifies the name that the log file will be saved as. The default is a generated filename that has the template `BL_TableNameUniquenumber`.

## BL\_PARALLEL= Table Option

Specifies whether or not to perform a parallel bulk load.

### Syntax

```
BL_PARALLEL=YES | NO
```

### Syntax Description

- **YES** — Specifies that a parallel load should be performed.
- **NO** — Specifies that a parallel load will not be performed. The default is NO.

## BL\_RECOVERABLE= Table Option

Specifies whether the LOAD process is recoverable.

### Syntax

```
BL_RECOVERABLE=YES | NO
```

### Syntax Description

- **YES** — Specifies that the LOAD process is recoverable. The default value is YES.
- **NO** — Specifies that the LOAD process is not recoverable. NO adds the UNRECOVERABLE keyword before the LOAD keyword in the control file.



## Details

To specify this option, you must first set BULKLOAD=YES.

Set this option to NO to improve direct load performance.



**Caution:** Be aware that an unrecoverable load does not log loaded data into the redo log file. Therefore, media recovery is disabled for the loaded table. For more information about the implications of using the UNRECOVERABLE parameter, see your Oracle utilities documentation.

## Example

The following example demonstrates the use of BL\_RECOVERABLE= to specify that the load is unrecoverable.

```
data x.recover_no (bulkload=yes bl_recoverable=no); cl=1; run;
```

For more information, see [BULKLOAD= Table Option](#).

## BL\_SKIP= Table Option

Specifies whether to skip the indicated number of rows before starting the bulk load.

### Syntax

```
BL_SKIP=n
```

### Syntax Description

- ***n*** — Specifies to skip the first *n* rows before loading. The default is 0, meaning no records are skipped.

## BL\_SKIP\_INDEX\_MAINTENANCE= Table Option

Specifies whether or not to perform index maintenance on the bulk load.

### Syntax

```
BL_SKIP_INDEX_MAINTENANCE=YES | NO
```

### Syntax Description

- **YES** — Specifies to stop index maintenance on the load. This will cause the index partitions that would have had index keys added to them to be marked *Index Unusable*. The index segment is inconsistent with the data it indexes. Index segments that are not affected by the load retain the *Index Unusable* state they had prior to the load.
- **NO** — Specifies that index maintenance will be performed on the load. The default is NO.

## BL\_SKIP\_UNUSABLE\_INDEXES= Table Option

Specifies whether to skip index entries that are in an unusable state and continue with the bulk load.

### Syntax

```
BL_SKIP_UNUSABLE_INDEXES=YES | NO
```

### Syntax Description

- **YES** — Indicates that the unusable index entry should be skipped. The default value is YES.
- **NO** — Indicates that the unusable index entry should not be skipped.

### Details

If an index in an Index Unusable state is encountered, by default, it is skipped and the load operation continues. This allows the SQL\*Loader to load a table with indexes that are in an unusable state prior to the beginning of the load. Indexes that are not in an unusable state at load time will be maintained by the SQL\*Loader. Indexes that are in an unusable state at load time will not be maintained but will remain in an unusable state at load completion.

If this bulk load option is not specified, the default value is specified in the Oracle database configuration parameter, SKIP\_UNUSABLE\_INDEXES. This value is specified in the initialization parameter file. The BL\_SKIP\_UNUSABLE\_INDEXES bulk load option overrides the value of the SKIP\_UNUSABLE\_INDEXES configuration parameter in the initialization parameter file.

## BULKLOAD= Table Option

Loads rows of data as one unit.

### Syntax

```
BULKLOAD=YES | NO
```

### Syntax Description

- **YES** — Calls a DBMS-specific bulk load facility in order to insert or append rows to a DBMS table.
- **NO** — Does not call the DBMS-specific bulk load facility. This is the default value.

### Details

Using BULKLOAD=YES is the fastest way to insert rows into a DBMS table.

# Teradata Reference

This chapter provides functionality details and guidelines for the Teradata® database that is supported by the DataFlux® Driver for Teradata (Driver for Teradata).

The following topics provide additional information about the Driver for Teradata:

- [Understanding the Driver for Teradata](#)
- [Database Functionality Support for Teradata](#)
- [Data Service Connection Arguments for Teradata](#)
- [Data Types for Teradata](#)
- [Naming Conventions for Teradata](#)
- [TSSQL Statement Table Options](#)

## Understanding the Driver for Teradata

The Driver for Teradata provides read and update access to Teradata database tables and creates tables that can be accessed by both the Federation Server and Teradata.

The Driver for Teradata supports most of the TSSQL functionality. The driver also supports an application's ability to submit native Teradata SQL statements.

The Driver for Teradata is a remote driver, which means that it connects to a server process to access data. The process might be running on the same machine as the Federation Server, or it might be running on another machine in the network.

## Database Functionality Support for Teradata

The following table lists the database functionality support level for Teradata that is provided by the Driver for Teradata. For general information about database functionality, see [Database Functionality Supported by Data Services](#).

Functionality	Description	Support	Comment
bookmark	Stored location used for quick retrieval and scrollable cursors	Supported	The table must have a primary key.
bulk operations	Ability to add, delete, or update a large quantity of data with one operation	Supports additions, deletions, and updates	
cursor support	Functionality that keeps track of the position in a result set and enables multiple operations on the result set	Static cursor	Cursor functionality is supported by the driver or by specifying that the Federation Server provides additional cursor support.
index	Component of a database that contains information		

Functionality	Description	Support	Comment
	about the physical location of data stored in a file		
integrity constraint	Functionality that ensures correctness, completeness, and compliance of data	Supported	
multi-row fetch	Enables an application to request more than one row at a time	Supported	
stored routines	SQL programs that perform computations or other tasks consistently	Stored procedures with parameters	
threaded I/O processing	Efficient method of processing a query with multiple threads	Not supported	
transactions	Process of protecting data by ensuring the integrity of data during multi-user update sessions	Supported	

## Data Service Connection Arguments for Teradata

To access data that is hosted on the Federation Server, a client must submit a connection string, which defines how to connect to the data. The data service connection arguments for a Teradata database include connection options and advanced options.

### Connection Options

Connection options are used to establish a connection to a data service. For each connection to the specified data service, one or more connection options should be specified in the create data service data definition language (DDL) statement. For more information on this DDL statement, see [Create Data Service](#).

The following connection options are supported for a Teradata database:

- **CATALOG=catalog-identifier;** — Specifies an arbitrary identifier for an SQL catalog, which groups logically related schemas. Any identifier is valid (for example, catalog=tera).

**Requirement:** You must specify a catalog.

- **DATABASE=database-name;** — Specifies the Teradata database. If you do not specify DATABASE=, you connect to the default Teradata database, which is often named the same as your user ID. If the database value that you specify contains spaces or non-alphanumeric characters, you must enclose it in quotation marks.
- **DRIVER=TERA;** — Identifies the data service to which you want to connect, which is a Teradata database.

**Requirement:** You must specify the driver.

- **SERVER=;** — Specifies the Teradata server identifier.
- **TDPID=dbname;** — Specifies a required connection option if you run more than one Teradata server. TDPID= operates differently for network-attached and channel-attached systems, as described below.

## Advanced Options

Advanced driver options are additional options that are not required in order to connect to the data source. They are used to establish connections to catalogs, data source names (DSNs), and schemas. While advanced options can also be used when connecting to a data service, doing so will cause the specified options to apply to all data service connections.

The following advanced options are supported for a Teradata database:

- **ACCOUNT=account-ID;** — Specifies an optional account number that you want to charge for the Teradata session.
- **DRIVER\_TRACE='API|SQL|ALL';** — Requests tracing information, which logs transaction records to an external file that can be used for debugging purposes. The Federation Server driver writes a record of each command that is sent to the database to the trace log based on the specified tracing level, which determines the type of tracing information. The tracing levels are as follows:
  - **ALL** — Activates all trace levels.
  - **API** — Specifies that API method calls be sent to the trace log. This option is most useful if you are having a problem and need to send a trace log to DataFlux® Technical Support for troubleshooting.
  - **DRIVER** — Specifies that driver-specific information be sent to the trace log.
  - **SQL** — Specifies that the SQL statements, such as SELECT and COMMIT, that are sent to the database be sent to the trace log.

**Default:** Tracing is not activated.

**Requirement:** If you activate tracing, you must also specify the location of the trace log with DRIVER\_TRACEFILE=. (Optional) You can control trace log formatting with DRIVER\_TRACEOPTIONS=.

**Interaction:** You can specify one trace level, or you can concatenate more than one by including the | (OR) symbol. For example: driver\_trace='api|sql' specifies to generate tracing information for API calls and SQL statements.

- **DRIVER\_TRACEFILE='filename';** — Specifies where to write trace information. The trace log is a text file. Include the filename and extension in single or double quotation marks. For example:

```
driver_tracefile='\\mytrace.log'
```

**Default:** There is no default trace log location.

**Requirement:** You must specify a location for a trace log if you activate tracing with DRIVER\_TRACE=.

**Interaction:** (Optional) You can control trace log formatting with `DRIVER_TRACEOPTIONS=`.

- **DRIVER\_TRACEOPTIONS=APPEND|THREADSTAMP|TIMESTAMP;** — Specifies options in order to control formatting and other properties for the trace log:
  - **APPEND** — Adds trace information to the end of an existing trace log. The contents of the file are not overwritten.
  - **THREADSTAMP** — Prepends each line of the trace log with a thread identification.
  - **TIMESTAMP** — Prepends each line of the trace log with a time stamp.

**Default:** The trace log is overwritten with no thread identification or time stamp.

- **PASSWORD=password;** — Specifies a Teradata password. The password you specify must be correct for your `USER=` value.

**Requirement:** You must specify the `PASSWORD=` option.

**Alias:** `PWD=`

- **Role=;** — Specifies the current role for the session.
- **USER=userid;** — Specifies a Teradata user ID. If the ID contains blanks or national characters, enclose it in quotation marks.`id`.

**Requirement:** You must specify the `USER=` option.

**Alias:** `UID=`

## Data Types for Teradata

When an application submits the Federation Server language statements, only the Federation Server data types are supported for column definitions. Native Teradata database data types that are not supported by the Federation Server are mapped to similar Federation Server data types for data retrieval. For information about the Federation Server data types, see [Data Service Data Types](#).

When an application submits Teradata SQL statements, native Teradata database data types are supported. Submitting Teradata SQL statements is available only to applications that consume the JDBC, ODBC, or OLE DB client-side drivers. For information about the native Teradata database SQL data types, see your Teradata documentation.

The following table lists the data types that are supported for Teradata:

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
BIGINT	BIGINT	Specifies a large signed, exact whole number with a precision of	BIGINT

Federation Server Language Keyword	Data Service Type Name	Data Service Type Definition	Federation Server Type Returned
		19.	
*	BLOB	Specifies a large binary object.	*
*	BYTE( <i>n</i> )	Specifies a fixed-length binary string from 1 to 64,000 bytes.	*
*	BYTEINT	Specifies a single-byte signed binary integer from -128 to 127.	*
CHAR( <i>n</i> )	CHAR( <i>n</i> )	Specifies a fixed-length character string from 1 to 64,000 bytes; the default is 1.	CHAR( <i>n</i> )
*	CLOB	Specifies a large character object	*
DATE	DATE	Specifies a date value in the range of 0001-01-01 through 9999-12-31. The format is YYYY-MM-DD.	DATE
*	DECIMAL( <i>p,s</i> )	Specifies a signed, fixed-point decimal number. The precision range is 1 to 18 and the scale default is 0.	*
DOUBLE	FLOAT	Specifies a signed, double precision, floating-point number with a precision of 53.	DOUBLE
INTEGER	INTEGER	Specifies a regular signed, exact whole number with a precision of 10.	INTEGER
*	LONG VARCHAR	Specifies a varying-length character string from 1 to 64,000 bytes.	*
SMALLINT	SMALLINT	Specifies a small signed, exact whole number with a precision of 5.	SMALLINT
TIME( <i>p</i> )	TIME( <i>p</i> )	Specifies a time value from 00:00:00.000000 to 23:59:61.999999 in the format HH:MM:SS.S(6).	TIME( <i>p</i> )
TIMESTAMP( <i>p</i> )	TIMESTAMP( <i>p</i> )	Specifies a date, time, and optional microsecond value in the range from 0001-01-01 00:00:00.000000 to 9999-12-31 23:59:61.999999 in the format YYYY-MM-DD HH:MM:SS.S(6).	TIMESTAMP( <i>p</i> )

<b>Federation Server Language Keyword</b>	<b>Data Service Type Name</b>	<b>Data Service Type Definition</b>	<b>Federation Server Type Returned</b>
TINYINT	TINYINT	Specifies a very small signed, exact whole number with a precision of 3.	TINYINT
VARBINARY( <i>n</i> )	VARBYTE( <i>n</i> )	Specifies a varying-length binary string from 1 to 64,000 bytes.	VARBINARY( <i>n</i> )
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Specifies a varying-length character string from 1 to 64,000 bytes.	VARCHAR( <i>n</i> )

\* For the Federation Server languages, this data type cannot be defined, and when data is retrieved, the Federation Server maps the native Teradata database data type to a similar Federation Server data type if possible.

## Naming Conventions for Teradata

The data objects that you can name in Teradata include tables, views, columns, indexes and macros. When naming a Teradata object, use the following conventions:

- A name must start with a letter unless you enclose it in double quotation marks.
- A name must be from 1 to 30 characters long.
- A name can contain the letters A through Z, the digits 0 through 9, the underscore (\_), \$, and #. A name in double quotation marks can contain any characters except double quotation marks.
- A name, even when enclosed in double quotation marks, is not case sensitive. For example, CUSTOMER is the same as customer.
- A name cannot be a Teradata reserved word.
- The name must be unique between objects. That is, a view and table in the same database cannot have the identical name.

## TSSQL Statement Table Options

Federation Server TSSQL (TSSQL) statement table options specify actions that apply only to the table with which they appear. They enable you to perform operations such as specifying a password for the driver file and renaming columns.

You can specify table options on a TSSQL statement on which you specify a table name, such as the CREATE TABLE, ALTER TABLE, or SELECT statement.

For more information on how to use these options, see "Using TSSQL Statement Table Options" in the *DataFlux Federation Server TSSQL Reference Guide*.



## Syntax

Specify a TSSQL statement table option immediately after the table name, within curly braces { } and including the keyword OPTION. To specify several table options, separate them with spaces or commas.

{OPTION *option-1=value* [... *option-n=value*] }



**Important:** While specifying the syntax for table options, you cannot have a space between the left curly brace { and the keyword OPTION. A space will result in a syntax error.

These examples show table options in TSSQL statements:

```
create table salary {option encrypt=yes read=green};
select * from salary {option read=green};
```

The following statement table options are supported by the Driver for Teradata.

- [TD\\_BUFFER\\_MODE](#)
- [TD\\_DATA\\_ENCRYPTION](#)
- [TD\\_ERROR\\_LIMIT](#)
- [TD\\_ERROR\\_TABLE\\_1](#)
- [TD\\_ERROR\\_TABLE\\_2](#)
- [TD\\_LOG\\_TABLE](#)
- [TD\\_LOGON\\_MECH](#)
- [TD\\_LOGON\\_MECH\\_DATA](#)
- [TD\\_MAX\\_SESSIONS](#)
- [TD\\_MIN\\_SESSIONS](#)
- [TD\\_TENACITY\\_HOURS](#)
- [TD\\_TENACITY\\_SLEEP](#)
- [TD\\_TRACE\\_LEVEL](#) and [TD\\_TRACE\\_LEVEL\\_INF](#)
- [TD\\_TRACE\\_OUTPUT](#)

## TD\_BUFFER\_MODE= Table Option

Specifies which type of load method is used.

## Syntax

```
TD_BUFFER_MODE= YES|NO
```

## Syntax Description

- **YES** — Must be set to YES in order to enable the bulk load feature.

- **NO** — This is the default value.

## TD\_DATA\_ENCRYPTION= Table Option

Specifies whether to activate full security encryption.

### Syntax

```
TD_DATA_ENCRYPTION=ON | OFF
```

### Syntax Description

- **ON** — Encrypts all SQL requests, responses, and data.
- **OFF** — No encryption occurs. This is the default setting.

## TD\_ERROR\_LIMIT= Table Option

Specifies the maximum number of records stored.

### Syntax

```
TD_ERROR_LIMIT=number of records
```

### Syntax Description

- ***number of records*** — Specifies the maximum number of records that can be stored in an error table before the Update driver job is terminated. The ErrorLimit specification applies to each instance of the Update driver.

### Details

The ErrorLimit specification must be greater than zero.

Specifying an invalid value terminates the Update driver. By default, ErrorLimit value is unlimited.

## TD\_ERROR\_TABLE\_1= Table Option

Specifies the name of the first error table.

### Syntax

```
TD_ERROR_TABLE_1= name of table
```

### Syntax Description

- ***name of table*** — Specifies the name of the error table which must be a new table name. The name of an existing table cannot be used, unless you are restarting a paused Update driver job. The default name for ErrorTable1 is tname\_ET.

### Details

ErrorTable1 contains records that were rejected during the acquisition phase of the Update driver job because of:

- Data conversion errors
- Constraint violations
- AMP configuration changes

For more information on the error table format and the procedure to correct errors, refer to the *Teradata FastLoad Reference*.

## TD\_ERROR\_TABLE\_2= Table Option

Specifies the name of the second error table.

### Syntax

```
TD_ERROR_TABLE_1= name of table
```

### Syntax Description

- **name of table** — Specifies the name of the error table which must be a new table name. The name of an existing table cannot be used, unless you are restarting a paused Update driver job. The default name for ErrorTable2 is ttname\_UV.

### Details

ErrorTable2 contains records that violated the unique primary index constraint. This type of error occurs during the application phase of the Update driver job.

For more information on the error table format and the procedure to correct errors, refer to the *Teradata FastLoad Reference*.

## TD\_LOG\_TABLE= Table Option

Specifies the name of the restart log table.

### Syntax

```
TD_LOG_TABLE=name of table
```

### Syntax Description

- **name of table** — Specifies the name of the restart log table, which contains the restart information.

### Details

If the restart log table name is not fully qualified, it is created under the use's default (logon) database. Alternately, a working database can be specified using the TD\_WORKINGDATABASE attribute. If the TD\_WORKINGDATABASE attribute is used, the restart log table name must be fully qualified, even if the restart log table is going to reside in the default (logon) database.

## TD\_LOGON\_MECH= Table Option

Specifies the logon mechanism.

## Syntax

```
TD_LOGON_MECH=mechanism
```

## Syntax Description

- ***mechanism*** — Specifies which logon mechanism is used. The job terminates if the attribute exceeds eight bytes.

## TD\_LOGON\_MECH\_DATA= Table Option

Specifies additional data.

## Syntax

```
TD_LOGON_MECH_DATA=mechanism
```

## Syntax Description

- ***mechanism*** — Specifies additional logon mechanism information.

## TD\_MAX\_SESSIONS= Table Option

Specifies the maximum number logon sessions.

## Syntax

```
TD_MAX_SESSIONS=maximum number
```

## Syntax Description

- ***maximum number*** — Specifies the maximum number of sessions to log on. The default is one session per available AMP. The maximum value cannot be more than the number of AMPS available.

## Details

The MaxSessions value must be greater than zero. Specifying a value less than one causes the job to terminate.

## TD\_MIN\_SESSIONS= Table Option

Specifies the minimum number of sessions for the Update driver job to run.

## Syntax

```
TD_MIN_SESSIONS=minimum number
```

## Syntax Description

- ***minimum number*** — Specifies the minimum number of sessions required for the Update driver job to continue. The default is one session.

## Details

The MinSessions value must be greater than zero and less than or equal to the maximum number of Update driver sessions. Specifying a value less than one will cause the Update driver to terminate.

## TD\_TENACITY\_HOURS= Table Option

Specifies the amount of time the Update driver continues trying to log on to the Teradata database.

### Syntax

```
TD_TENACITY_HOURS=number of hours
```

### Syntax Description

- ***number of hours*** — Specifies how long (in hours) the Update driver will attempt to log on to the Teradata database, when the maximum number of load and export operations are already running. The default value is four hours.

## Details

The following conditions exist:

- Specifying a value of zero will disable the tenacity feature.
- To enable the tenacity feature, the hours value must be greater than zero.
- Specifying a value of less than zero terminates the Update driver.

## TD\_TENACITY\_SLEEP= Table Option

Specifies the amount of time the Update driver pauses, before retrying to log on to the Teradata database.

### Syntax

```
TD_TENACITY_SLEEP=number of minutes
```

### Syntax Description

- ***number of minutes*** — Specifies the number of minutes the Update driver pauses before retrying to log on when the maximum number of load and export operations are already running on the Teradata Database. The default is six minutes.

## Details

The minutes value must be greater than zero. If you specify a value less than one, the Update driver responds with an error message and terminates the job.

## TD\_TRACE\_LEVEL= and TD\_TRACE\_LEVEL\_INF= Table Options

These two options can both be used to specify the trace levels for driver tracing. TD\_TRACE\_LEVEL sets the primary trace level, while TD\_TRACE\_LEVEL\_INF sets the secondary trace level.

### Syntax

```
TD_TRACE_LEVEL=trace level  
TD_TRACE_LEVEL_INF=trace level
```

### Syntax Description

- ***trace level*** — Specifies the types of diagnostic messages written by each instance of the driver to an external log file. The diagnostic trace function provides more detailed information (including the version number) in the log file to aid in problem tracking and diagnosis. The following trace levels are available:
  - **TD\_OFF** — Disables driver tracing. TD\_OFF is the default setting for both driver tracing and infrastructure tracing. No external log file is produced unless this default is changed. Specifying TD\_OFF for both driver tracing and infrastructure tracing is the same as disabling tracing.
  - **TD\_OPER** — Activates the tracing function for driver specific activities. The absence of any value for the PauseAcq attribute means that the Update driver job will execute both the acquisition phase and the application phase without pausing. This will distribute all of the rows that were sent to the Teradata database during the acquisition phase to their final destination on the AMPs.
  - **TD\_OPER\_CLI** — Activates the tracing function for CLIV2-related activities (interaction with the Teradata database).
  - **TD\_OPER\_NOTIFY** — Activates the tracing function for activities related to the Notify feature.
  - **TD\_OPER\_OPCOMMON** — Activates the tracing function for activities involving the opcommon library.
  - **TD\_OPER\_ALL** — Activates the tracing function for all of the tracing activities.

### Details

If the TraceLevel is set to any value other than TD\_OFF, an external log file is created for each instance of the driver.

The trace levels for infrastructure tracing should only be used when you are directed to by Teradata support. TD\_OFF, which disables infrastructure tracing, should always be specified.

## TD\_TRACE\_OUTPUT= Table Option

Specifies the name of the external file used for trace messages.

### Syntax

```
TD_TRACE_OUTPUT=filename
```

## Syntax Description

- *filename* — Creates a new file name using the name of the driver, followed by a time stamp.



**Note:** If a file with the specified name already exists, then the existing file is overwritten.

# Technical Support

This section addresses questions and issues related to DataFlux Federation Server. If you do not find your answer, please contact [DataFlux Technical Support](#).

## Best Practices

### The SYSTEM user should create and define the Federation Server administrators

#### Summary

When using the Federation Server Manager, you should use the SYSTEM user to create and define one or more Federation Server administrators. Then, all Federation Server configuration and administration should be performed by the non-SYSTEM (Administrator) users.

#### Details

The Federation Server supports SYSTEM users, which are one or more users that are unconditionally and implicitly granted all privileges. A SYSTEM user cannot be denied privileges on any object. The SYSTEM user is required to bootstrap the system, and is defined through the Federation Server configuration file.

The SYSTEM user should identify users who will be administrators of the Federation Server, and make them administrators by granting them the Administer privilege on the server object. Like SYSTEM users, administrators are unconditionally and implicitly granted all privileges on the Federation Server. However, if these users are revoked their Administer privilege, then they become standard users that have privileges granted to and denied from them.

The Federation Server Manager will always grant and deny privileges to users using the AS ADMINISTRATOR clause, such that privileges are recorded with the ADMINISTRATOR role as grantor. Should the user acting as administrator later be revoked of Administer privilege or even removed from the system, the privilege record will still be valid and persisted.

If a Data Source Name (DSN) is created by either the SYSTEM user or an Administrator, the DSN is created using the AS ADMINISTRATOR clause, which means that the ADMINISTRATOR role owns the DSN, not the individual creating it. Therefore, if the administrator user is later removed from the system, the DSN will not be deleted with the user. In future releases, the Federation Server may extend its use of roles and ownership.



**Note:** If a DSN is created by a user other than the SYSTEM user or Administrator, the DSN is owned by the individual user. If that user is later removed from the system, DSN ownership should be transferred to another user. For more information, see [Orphaned Accounts](#).

### Placing operating system protections on the Federation Server installation directory

The recommended operating system protections for Windows are listed in the following table:



Directories	Users	Default Permissions
<i>DataFlux-configuration-directory</i> <i>DataFlux-configuration-directory\Federation Server</i>	Installer, Administrator	Full Control
	Process user	Read and Execute, List Folder Contents
<i>DataFlux-configuration-directory\Federation Server\version\var</i>	Installer, Administrator	Full Control
	Process user	Read, Write, List Folder Contents
	The user who backs up the Federation Server; Backup Administrator	Read, List Folder Contents
<i>TranPath as specified in the server configuration file</i>	Installer, Administrator	Full Control
	Process user	Read, Write, List Folder Contents
	The user who backs up the Federation Server; Backup Administrator	Read, List Folder Contents

 **Note:** All other users have no access.

The recommended operating system protections for UNIX® are listed in the following table:

Directories	Users	Default Permissions
<i>DataFlux-configuration-directory</i> <i>DataFlux-configuration-directory\Federation Server</i>	Installer, Administrator	Read, Write, Execute
	Process user	Read, Execute
<i>DataFlux-configuration-directory\Federation Server\version\var</i>	Installer, Administrator	Read, Write, Execute
	Process user	Read, Write, Execute
	The user who backs up the Federation Server; Backup Administrator	Read, Execute
<i>TranPath as specified in the server configuration file</i>	Installer, Administrator	Read, Write, Execute
	Process user	Read, Write, Execute
	The user who backs up the Federation Server; Backup Administrator	Read, Execute

 **Note:** All other users have no access.

### Enabling Federation Server Authorization Enforcement Security

It is recommended that you enable SQL authorization enforcement for control of:

1. SQL command sets that individual users and groups are authorized to execute, and
2. sets of data services, catalogs and schemas on which these commands are authorized.

Federation Server uses a tiered approach for SQL authorization enforcement which limits the effective privilege/permission level of Database Management System (DBMS) accounts used to establish back-end database connections held in a server connection. First, SQL statements are authorized in the server tier before they are passed to the back-end DBMS account tier where SQL authorization enforcement takes place for the database user.

Federation Server users who already know credentials of back-end DBMS accounts can simply connect directly to them and operate on all visible SQL data at the privilege/permission level of a DBMS user, effectively bypassing any security policies configured in the Federation Server authorization tier. For this reason, it is best practice to couple the use of shared logins with security enabled connections. Shared SQL data can be secured for multiple Federation Server users using the undisclosed credentials of a single back-end DBMS account acquired on behalf of the user by the server.

## Troubleshooting the DataFlux Federation Server Deployment

This chapter provides information and techniques to use while attempting to identify and resolve problems related to the Federation Server deployment. To troubleshoot your installation and configuration, use the logging facility and tracing.

### Logging Facility

The initial Federation Server installation and configuration includes the SAS® logging facility. The logging facility is a flexible, configurable framework for collecting, categorizing, and filtering events that are generated by DataFlux processes and for writing events to a variety of output devices.

To configure the logging facility, the Federation Server provides a default logging configuration file named `dfs_log.xml` that controls the destination, contents, and format of the log. The configuration file specifies options for the logging facility components, which include loggers, appenders, levels, filters, and the layout of log messages. The `dfs_log.xml` configuration file specifies the path and filename of the default logging configuration file.

For more information about the Federation Server configuration files, see [Managing Configuration Files and Scripts](#).

### Initial Logging Configuration for the Federation Server

The default logging facility configuration for the Federation Server includes a definition for the RollingFileAppender. The appender routes events to a rolling log file. The rolling log file is configured as follows:

- A new log is created when the date changes and when a new server process is started.
- Events are written by using a layout that includes the current date, current time, logging level, process ID, the user identity that is associated with the event, and a message.

- The name of the rolling log file follows the following convention:

```
dfs_%d_%S{pid}.log
```

where %d is the date and %S{pid} is the process ID number (PID) for the Federation Server.

- The rolling log files are placed in the `/var/log` directory.
- When a new rolling log file is created, a heading is written to the file. The heading identifies the server's host machine, operating system, operating system release number, DataFlux release number, and server start-up command.

The following table lists the loggers that reference the RollingFileAppender:

Logger Name	Logging Level
Admin	ERROR
App	INFO
Audit	INFO
IOM	INFO

## Modifying the Server Logging Configuration

You can modify the logging facility configuration for the Federation Server by modifying the `dfs_log.xml` file. Before modifying the file, be sure to make a backup copy. Here are some examples of changes that you might want to make:

- configure RollingFileAppender to use a different log filename, to roll over log files more or less frequently, or to roll over log files based on file size rather than date
- specify additional appenders
- use filters to limit the events that are written to an appender
- configure a different message layout for an appender

## Trace Log

By tracing each internal API routine that is called by the application, a trace log records transactions that can be used for debugging connection and processing issues. For example, you can request information that traces the TSSQL statements that are submitted to a data service.



**Note:** By default, tracing is not activated. It is recommended that you not activate tracing unless you are instructed to do so by DataFlux Technical Support.

Tracing can be activated by using the following methods:

- connection string options
- server start-up options
- data service connection arguments

- DataFlux system options

When you activate tracing, you also specify the physical location where the transaction records are saved. Because the Federation Server supports one root file trace log directory and multiple subdirectories, you can group trace logs if necessary.

## **Failure to Authenticate**

If you experience difficulty authenticating to the Federation Server, see the *DataFlux Authentication Server Administrator's Guide* for troubleshooting information or contact the Authentication Server administrator.

# Appendix A: TSSQL DDL Statements

In this appendix, PostgreSQL syntax as well as proposed SQL 99 syntax for certain SQL/MED (Management of External Data) features is used. The following conventions are used in the command syntax, in addition to those specified in [Conventions Used In This Document](#):

- Brackets ( [ and ] ) indicate optional parts of the syntax.
- Braces ( { and } ) and vertical lines ( | ) indicate that you must choose one alternative.
- Dots ( . . . ) indicate that the preceding element can be repeated.

Data Definition Language (DDL) statements can be used to perform the following tasks on Federation Server objects:

- [GENERIC OPTIONS Syntax](#)
- [ALTER GENERIC OPTIONS Syntax](#)
- [ALTER SERVER Statement](#)
- [CREATE DATA SERVICE Statement](#)
- [DROP DATA SERVICE Statement](#)
- [ALTER DATA SERVICE Statement](#)
- [CREATE CATALOG Statement](#)
- [DROP CATALOG Statement](#)
- [ALTER CATALOG Statement](#)
- [CREATE SCHEMA Statement](#)
- [DROP SCHEMA Statement](#)
- [ALTER SCHEMA Statement](#)
- [CREATE DSN Statement](#)
- [DROP DSN Statement](#)
- [ALTER DSN Statement](#)
- [DROP AUTHID Statement](#)

Once the objects have been created, you can set grant, deny, and revoke privileges. The following topics contain the syntax for administering security policies using Federation Server TSSQL (TSSQL):

- [GRANT and DENY Statements](#)
- [REVOKE Statements](#)

# GENERIC OPTIONS Syntax

A set of comma separated name or name-value pairs, within an OPTIONS list.

## Syntax

```
generic-options ::=
    "{" OPTIONS ["("] generic-option-list [")"] "}"

generic-option-list ::=
    generic-option [ {"," generic-option} ... ]

generic-option ::=
    option-name [ {option-value | option-list} ]

option-list ::=
    "(" option-value [ {"," option-value} ... ] ")"

option-value ::=
    quoted-identifier
```

# ALTER GENERIC OPTIONS Syntax

A set of comma separated name or name-value pairs with optional operation keywords, within an OPTIONS list.

## Syntax

```
alter-generic-options ::=
    "{" OPTIONS ["("] alter-generic-option-list [")"] "}"

alter-generic-option-list ::=
    alter-generic-option [ {"," alter-generic-option} ... ]

alter-generic-option ::=
    [ alter-operation ] option-name
    [ { option-value | option-list } ]

alter-operation ::=
    ADD | SET | XSET | DROP
```



**Note:** If a value is omitted for *alter-operation*, the default operation is ADD.

## Syntax Description

The possible values for the *alter-operation* option are:

- **ADD** — Adds the option.
- **SET** — Changes the option that already exists.
- **XSET** — Sets the option, if it has already been added. Or, adds the option if it does not already exist.
- **DROP** — Drops the option.

# ALTER SERVER Statement

Allows you to change the server configuration by specifying a trace file path, shared login key, shared login manager, and shared login password. The location of the tracefilepaths is relative to the ContentRoot value that is specified in the Federation Server configuration file, which by default is `dfs_serv.xml`. The default trace file path is `drive:\Program Files\DataFlux\FederationServer\version\var`.

## Syntax

```
ALTER SERVER
[ alter-server-options ]
```

## Syntax Description

- **alter-server-options** — Specifies the list of server options to alter. This includes the TraceFilePath, SharedLoginKey, SharedLoginManager, and SharedLoginPassword options.

```
alter-server-options ::=
    "{" OPTIONS ["("] alter-server-option
    [ { "," alter-server-option } ... ] [")"] "}"
```

- **alter-server-option** — Specifies the server option to alter.

```
alter-server-option ::=
    [ alter-operation ] server-option
```

- **server-option** — Specifies the server configuration options as:
  - **TRACEFILEPATH | TRACEFILE\_ROOT** *directory-path* — Specifies the trace file content location root for TRACEFILE keywords and the TRACEFILE environment handle attribute. The trace file path is used to store all trace files that are created, either from Federation Server start-up, or when enabled on a connection. This path may be absolute or relative to the server's content root location that is specified in the server configuration file.
  - **SHAREDLOGINMANAGER** *manager* — The Authentication Server manager account that will be used to retrieve the shared login map credentials.
  - **SHAREDLOGINPASSWORD** *password* — The Authentication Server password for the manager account that will be used to retrieve the shared login map credentials.
  - **SHAREDLOGINKEY** *key* — The Authentication Server grouping key used to search for shared login map credentials.



**Note:** The Shared Login password is encrypted. By default, SASProprietary encryption is used; however, the method of encryption can vary depending on the security platform where Federation Server is installed. If DataFlux Secure for Federation Server has been installed, AES encryption can be used.

## Examples

```
ALTER SERVER {OPTIONS add TRACEFILEPATH "C:\tracefiles"}
```

```
ALTER SERVER {OPTIONS add TRACEFILEPATH "logs\tracefiles"}
```

```
ALTER SERVER {OPTIONS xset SHAREDLOGINKEY 'DefaultKey'}
```

# CREATE DATA SERVICE Statement

Allows you to create a data service for a data source.

## Syntax

```
CREATE [DATA] SERVICE data-service
TYPE data-service-type
[ VERSION data-service-version ]
[ CATALOG [NAME] catalog-name ]
[ DOMAIN [NAME] domain-name ]
[ data-service-options ]
```

## Syntax Description

- ***data-service*** — Specifies the data service name. The following naming rules apply:
  1. The specified name must not match the reserved name of the internal data service, BASE.
  2. The specified name must not match the name of any existing data service.
  3. The specified name must not match the name of any existing catalog for data services defined for data sources that do not support catalog names, unless the CATALOG option is specified with a different name.
- ***data-service-type*** — Specifies the data service type. For a more information on the supported data service data types, see [Supported Data Service Types](#).
- ***data-service-version*** — Specifies the quoted data service version number.
- ***catalog-name*** — Specifies the catalog (logical) name associated with the data service. Must be unique within the server. If omitted, for data sources that do not support catalogs, the default logical catalog name matches the data service name.
- ***domain-name*** — Specifies the Authentication Server domain name associated with this data service. If omitted, the default domain name matches the data service name.
- ***data-service-options*** — Specifies the list of data service options.

```
data-service-options ::=
  "{" OPTIONS ["(") data-service-option
    [ { "," data-service-option } ... ] [")"] "}"
```

- ***data-service-option*** — Specifies the data service option.

```
data-service-option ::=
  conopts-configuration-list
```

- ***conopts-configuration-list*** — If DRIVER *driver-name* is omitted, the default driver for the data service is assumed. Associated options within the CONOPTS list are used for connections using the appropriate driver. Some data services such as ORACLE accept connections from the DataFlux® Driver for ODBC (Driver for ODBC) as well. For these data services, two CONOPTS lists may be configured, one per driver to accept connections for the two drivers. The Driver for ODBC accepts a CONOPTS



driver connection string option. To configure this option and sub-options within it, the configuration option format would be CONOPTS( driver ODBC, conopts( ... ) ). The inner CONOPTS option happens to be a specific driver connection string (list-valued) option while the outer CONOPTS groups arbitrary driver connection string options configured for the service.

```
conopts-configuration-list ::=
CONOPTS "(" [DRIVER driver-name] ["," driver-connection-string-
option ...] ")" ...
```

- **driver-name** — Specifies the driver name. For a list of the default drivers for each data service, see [Supported Data Service Types](#).
- **driver-connection-string-option** — Specifies the connection options that correspond to the driver which is specified in DRIVER *driver-name*. The DATA\_SERVICE and CATALOG connection string options should not be specified here since they are implied by the data service and its configured catalogs.

While both connection options and advanced options are valid to use as driver connection string options, typically, only connection options are used. For more information on which connection options and advanced options are supported for each data service, see the data service reference chapter for your driver. For information specific to the data service you are configuring, go to the section that corresponds to your data service.

## Examples

```
CREATE SERVICE ORASERV TYPE ORACLE domain ORA1 {OPTIONS ( conopts ( Driver
odbc, conopts(DSN tktsora)), conopts ( Driver oracle, PATH tktsora) ) }
```

```
CREATE DATA SERVICE SQLServer1 TYPE SQLSVR domain SQLSERVER {OPTIONS ( conopts
( conopts(DSN tktssql)) ) }
```

```
CREATE SERVICE DB2_SERVICE TYPE DB2UNXPC VERSION '1.0.0' domain DB2 {OPTIONS (
conopts (DB DEV1) ) }
```

```
CREATE SERVICE TERA_SERVICE TYPE TERADATA VERSION '1.0.0' domain TERA {OPTIONS
( conopts (Server kaching.unx.df.com) ) }
```

```
CREATE SERVICE MYSQL_SERVICE TYPE MYSQL VERSION '1.0.0' domain MYSQL {OPTIONS (
conopts (Server tktsmyl.na.df.com, DB tktstst1) ) }
```

```
CREATE SERVICE SAPSERV TYPE SAP VERSION '1.0.0' DOMAIN SAPDOMAIN
{OPTIONS conopts(ashost sapsrv.sup.com, sysnr 03, batch 1)}
```

## DROP DATA SERVICE Statement

Allows you to drop a data service.

### Syntax

```
DROP [DATA] SERVICE data-service [ drop-disposition ]
```

## Syntax Description

- **data-service** — Specifies the data service name.
- **drop-disposition** — Specifies the drop disposition as one of the following values:

```
drop-disposition ::=
    {RESTRICT | CASCADE} [FORCE]
```

- **RESTRICT** — Specifies that the drop target is empty. This is the default value.
- **CASCADE** — Specifies that contained objects are dropped.
- **FORCE** — Specifies the optional FORCE keyword that will suppress error messages when the data service does not exist. This additional option does not affect the performance of the RESTRICT or CASCADE options.

## Examples

```
drop DATA SERVICE ORACLE3

drop service "MYSQL_SERVICE" cascade force

drop data service ORACLE1 cascade
```

## ALTER DATA SERVICE Statement

Allows you to change the name of a data service. You can also change common data service attributes such as version, catalog, and domain, and other data service specific options.

## Syntax

```
ALTER [DATA] SERVICE data-service RENAME TO newname

ALTER [DATA] SERVICE data-service
    [ VERSION data-service-version ]
    [ CATALOG [NAME] catalog-name ]
    [ DOMAIN [NAME] domain-name ]
    [ alter-data-service-options ]
```

## Syntax Description

- **data-service** — Specifies the data service name.
- **newname** — Specifies the new data service name.
- **data-service-version** — Specifies the data service version.
- **catalog-name** — Specifies the catalog name.
- **domain-name** — Specifies the domain name.
- **alter-data-service-options** — Specifies the list of data service options to alter.

```
alter-data-service-options ::=
    "{" OPTIONS ["(") alter-data-service-option
        [ { "," alter-data-service-option } ... ]
        [")"] "}"
```

- **alter-data-service-option** — Specifies the data service option to alter.

```
alter-data-service-option ::=
[ alter-operation ] data-service-option
```

- **data-service-option** — Specifies the data service option.

```
data-service-option ::=
conopts-configuration-list
```

- **conopts-configuration-list** — If DRIVER *driver-name* is omitted, the default driver for the data service is assumed. Associated options within the CONOPTS list are used for connections using the appropriate driver. Some data services such as ORACLE accept connections from the Driver for ODBC as well. For these data services, two CONOPTS lists may be configured, one per driver to accept connections for the two drivers. The Driver for ODBC accepts a CONOPTS driver connection string option. To configure this option and sub-options within it, the configuration option format would be CONOPTS( driver ODBC, conopts( ... ) ). The inner CONOPTS option happens to be a specific driver connection string (list-valued) option while the outer CONOPTS groups arbitrary driver connection string options configured for the service.

```
conopts-configuration-list ::=
CONOPTS "(" [DRIVER driver-name] ["," driver-connection-
string-option ...] ")" ...
```

- **driver-name** — Specifies the driver name. For a list of the default drivers for each data service, see [Supported Data Service Types](#).
- **driver-connection-string-option** — Specifies the connection options that correspond to the driver which is specified in DRIVER *driver-name*. The DATA\_SERVICE and CATALOG connection string options should not be specified here since they are implied by the data service and its configured catalogs.

While both connection options and advanced options are valid to use as driver connection string options, typically, only connection options are used. For more information on which connection options and advanced options are supported for each data service, see the data service reference chapter for your driver. Information specific to the data service you are configuring can be found in the section for your data service.

## Examples

```
ALTER DATA SERVICE ORACLE3 VERSION '1.1.1' {OPTIONS conopts(Driver odbc,
ODBC_DSN tktsora)}
```

```
ALTER DATA SERVICE ORACLE3 RENAME TO ORACLE3_RENAME
```

```
ALTER DATA SERVICE ORACLE3_RENAME {OPTIONS DROP conopts(driver odbc) }
```

```
alter service service1 version '2' catalog newcatalog
```

```
alter service ORACLE4 DOMAIN ORA8
```

```
ALTER DATA SERVICE ORACLE3_RENAME {OPTIONS SET conopts(driver odbc, ODBC_DSN
tktsora3) }
```

```
ALTER SERVICE SAPSERV {OPTIONS conopts(xset batch 1, xset destgroup DATAFLUX)};
```

## CREATE CATALOG Statement

Allows you to create a catalog.

### Syntax

```
CREATE CATALOG catalog UNDER data-service
  [ NATIVE NAME native-name ]
  [ create-catalog-options ]
```

### Syntax Description

- **catalog** — Specifies the catalog name.
- **data-service** — Specifies the data service name under which the catalog is to be created.
- **native-name** — Specifies the native catalog name. Specified when the native catalog name is not unique within the server. The native name should be used to resolve catalog name collisions between multiple data services that support catalogs. Client SQL always references the catalog via the logical catalog name (catalog) regardless of whether or not a native name is specified. Specifying a native name that matches the logical name does nothing.
- **create-catalog-options** — Specifies the options to create a catalog. This option only applies to the BASE data service.

```
create-catalog-options ::=
conopts-configuration-list
```

- **conopts-configuration-list** — If DRIVER *driver-name* is omitted, the default driver for the data service is assumed. Associated options within the CONOPTS list are used for connections using the appropriate driver. The multiple driver syntax is not supported.

```
conopts-configuration-list ::=
CONOPTS "(" [DRIVER driver-name] ["," driver-connection-string-
option ...] ")" ...
```

- **driver-name** — Specifies the driver name. For a list of the default drivers for each data service, see [Supported Data Service Types](#).
- **driver-connection-string-option** — Specifies the connection options that correspond to the driver which is specified in DRIVER *driver-name*.

Advanced options are valid to use as driver connection string options; connection options are invalid. For more information on which advanced options are supported for each data service, see the data service reference chapter for your driver. For information specific to the data service you are configuring, go to the section that corresponds to your data service.

### Examples

```
CREATE CATALOG "catalog1_BASE" UNDER BASE
```

```
CREATE CATALOG "TKTEST" UNDER SQLServer1

CREATE CATALOG "Catalog1" UNDER SQLServer1 NATIVE NAME "TKTEST"

CREATE CATALOG "c1" UNDER BASE {OPTIONS conopts (COMPRESS YES)}
```

## DROP CATALOG Statement

Allows you to drop a catalog.

### Syntax

```
DROP CATALOG catalog [ drop-disposition ]
```

### Syntax Description

- ***catalog*** — Specifies the catalog name.
- ***drop-disposition*** — Specifies the drop disposition and is one of the following values:

```
drop-disposition ::=
{RESTRICT | CASCADE} [FORCE]
```

- **RESTRICT** — Specifies that the drop target is empty. This is the default value.
- **CASCADE** — Specifies that contained objects are dropped.
- **FORCE** — Specifies the optional FORCE keyword that will suppress error messages when the data service does not exist. This additional option does not affect the performance of the RESTRICT or CASCADE options.

### Examples

```
drop CATALOG "Catalog3"

drop catalog "catalog1_BASE" cascade
```

## ALTER CATALOG Statement

Allows you to change the name of a catalog. You can also change the native catalog name and the advanced options that are driver-specific. For information on which advanced options are supported for each data service, see the data service reference chapter for your driver. Information specific to the data service you are configuring can be found in the section for your data service.

### Syntax

```
ALTER CATALOG catalog RENAME TO newcatalogname
ALTER CATALOG catalog
    [ NATIVE NAME native-name ]
    [ alter-catalog-options ]
```

### Syntax Description

- ***catalog*** — Specifies the catalog name.
- ***newcatalogname*** — Specifies the new catalog name.

- **native-name** — Specifies the name of the native catalog.
- **alter-catalog-options** — Specifies the options to alter the catalog. This option only applies to the BASE data service. The syntax for *alter-catalog-options* is the same as the syntax for *alter-generic-options*. For more information, see [ALTER GENERIC OPTIONS Syntax](#). All *create-catalog-options* are also supported. For more information, see [create-catalog-options](#).

## Examples

```
ALTER CATALOG "catalog3_BASE" RENAME TO "catalog3_BASE_RENAME"

ALTER CATALOG "Catalog3" NATIVE NAME "TKTEST3_RENAME"

ALTER CATALOG "catalog1_BASE" {OPTIONS add CONOPTS(DRIVER BASE, ACCESS
READONLY)}

ALTER CATALOG "catalog1_BASE" {OPTIONS set (CONOPTS(DRIVER BASE, ACCESS
READONLY))}

ALTER CATALOG "catalog1_BASE" {OPTIONS xset CONOPTS(DRIVER BASE, COMPRESS YES)}

ALTER CATALOG "catalog1_BASE" {OPTIONS drop CONOPTS(DRIVER BASE)}
```

## CREATE SCHEMA Statement

Allows you to create a schema.

### Syntax

```
CREATE SCHEMA [ catalog "." ] schema
[ create-schema-options ]
```

### Syntax Description

- **catalog** — Specifies the optional catalog name under which to create the schema. Useful for data services defined for data sources that support catalog names. For those that do not, the catalog name must be that logical catalog name which defaults to the name of the data service.
- **schema** — Specifies the schema name.
- **create-schema-options** — Specifies the options to create the schema.

```
create-schema-options ::=
"{" OPTIONS ["("] schema-option
[ { "," schema-option } ... ] [")"] "}"
```

- **schema-option** — Specifies the syntax for schema options. This option only applies to the BASE data service.

- **conopts-configuration-list** — If DRIVER *driver-name* is omitted, the default driver for the data service is assumed. Associated options within the CONOPTS list are used for connections using the appropriate driver.

```
conopts-configuration-list ::=
CONOPTS "(" [DRIVER driver-name] ["," driver-connection-string-
option ...] ")" ...
```

- **driver-name** — Specifies the driver name. For a list of the default drivers for each data service, see [Supported Data Service Types](#).
- **driver-connection-string-option** — Specifies the connection options that correspond to the driver which is specified in DRIVER *driver-name*.

Advanced options are valid to use as driver connection string options; connection options are invalid. For more information on which advanced options are supported for each data service, see the data service reference chapter for your driver. For information specific to the data service you are configuring, go to the section that corresponds to your data service.

- **PRIMARYPATH** *path* — Specifies the physical location for the SAS library, which is a collection of one or more SAS files. For example, in directory-based operating environments, a SAS library is a group of SAS files that are stored in the same directory. This option is required for BASE schemas.

```
path ::=
    quoted-identifier
```

- **quoted-identifier** — Specifies a single quoted or double quoted name.

## Examples

```
CREATE SCHEMA "catalog1_BASE"."schema1_BASE" {OPTIONS (primarypath
'C:\schema1_BASE')}
```

```
CREATE SCHEMA "ORACLE1"."TKTSTST1"
```

```
CREATE SCHEMA "catalog1"."schema1" {OPTIONS primarypath 'C:\my_schema', conopts
(LOCKTABLE EXCLUSIVE)}
```

## DROP SCHEMA Statement

Allows you to drop a schema.

### Syntax

```
DROP SCHEMA [ catalog "." ] schema [ drop-disposition ]
```

### Syntax Description

- **catalog** — Specifies the catalog name.
- **schema** — Specifies the data service name.
- **drop-disposition** — Specifies the drop disposition and is one of the following values:

```
drop-disposition ::=
{RESTRICT | CASCADE} [FORCE]
```

- **RESTRICT** — Specifies that the drop target is empty. This is the default value.
- **CASCADE** — Specifies that contained objects are dropped.

- **FORCE** — Specifies the optional FORCE keyword that will suppress error messages when the data service does not exist. This additional option does not affect the performance of the RESTRICT or CASCADE options.

## Examples

```
DROP SCHEMA "catalog1_BASE"."schema1_BASE"
```

```
DROP SCHEMA "catalog1_BASE"."schema1_BASE" force
```

## ALTER SCHEMA Statement

Allows you to change the name of a schema. You can also alter advanced options that are driver-specific.

### Syntax

```
ALTER SCHEMA [ catalog "." ] schema RENAME TO newschema
ALTER SCHEMA [ catalog "." ] schema
[ alter-schema-options ]
```

### Syntax Description

- **catalog** — Specifies the catalog name.
- **schema** — Specifies the schema name.
- **newschema** — Specifies the new schema name.
- **alter-schema-options** — Specifies the options to alter the schema.

```
alter-schema-options ::=
    "{" OPTIONS ["("] alter-schema-option
    [ {"", " alter-schema-option} ... ] [")"] "}"
```

- **alter-schema-option** — Specifies the schema option to alter. This option only applies to the BASE data service.

```
alter-schema-option ::=
    [ DROP schema-option-name ]
    [ {ADD | SET} schema-option ]
```

- **schema-option** — Specifies the syntax for schema options.
  - **conopts-configuration-list** — If DRIVER *driver-name* is omitted, the default driver for the data service is assumed. Associated options within the CONOPTS list are used for connections using the appropriate driver.
 

```
conopts-configuration-list ::=
CONOPTS "(" [DRIVER driver-name] [", " driver-
connection-string-option ...] ")" ...
```
  - **driver-name** — Specifies the driver name. For a list of the default drivers for each data service, see [Supported Data Service Types](#).
  - **driver-connection-string-option** — Specifies the connection options that correspond to the driver which is specified in DRIVER *driver-name*.



Advanced options are valid to use as driver connection string options; connection options are invalid. For more information on which advanced options are supported for each data service, see the data service reference chapter for your driver. For information specific to the data service you are configuring, go to the section that corresponds to your data service.

- **PRIMARYPATH** *path* — Specifies the physical location for the SAS library, which is a collection of one or more SAS files. For example, in directory-based operating environments, a SAS library is a group of SAS files that are stored in the same directory. This option is required for BASE schemas.

```
path ::=  
quoted-identifier
```

- **quoted-identifier** — Specifies a single quoted or double quoted name.

## Examples

```
ALTER SCHEMA "catalog1_BASE"."schema3_BASE" RENAME TO "schema3_BASE_RENAME"
```

```
ALTER SCHEMA "catalog1_BASE"."schema3_BASE" {OPTIONS set primarypath  
'C:\mydir'}
```

```
ALTER SCHEMA "catalog1_BASE"."schema3_BASE" {OPTIONS add conopts (LOCKTABLE  
SHARE) }
```

## CREATE DSN Statement

Allows you to create a data service level DSN.

### Syntax

```
CREATE DSN dsn-name UNDER data-service  
create-dsdsn-options
```

### Syntax Description

- **dsn-name** — Specifies the DSN name.
- **data-service** — Specifies the data service name. The following naming rules apply:
  1. The specified name must not match the reserved names of the internal BASE data service.
  2. The specified name must not match the name of any existing data service.
  3. The specified name must not match the name of any existing catalog for data services defined for data sources that do not support catalog names, unless the CATALOG option is specified with a different name.
- **create-dsdsn-options** — Specifies the data service-level DSN option as:

```
create-dsn-options
```

```
{NOPROMPT | CONNECT} connection-string-options
```

- **connection-string-options** — Specifies the connection string. CONNECT is the default.
- **create-dsn-options** — Specifies the common DSN option as one of the following:
  - **DESC[RIPTION]** *description-text* — Specifies the quoted description of the DSN.
  - **dsn-config-options** — Specifies the options to configure the DSN.

```
dsn-config-options ::=
  "{" OPTIONS ["("] dsn-config-option
                        [ {"", dsn-config-option} ... ]
                        [")"] "}"
```

- **dsn-config-option** — Specifies the DSN configuration option as one of the following:
  - **TS\_SQL Y[ES] | T[RUE] | 1 | N[O] | F[ALSE] | 0** — Specifies whether or not to use TSSQL dialect. Base DSN connections always use TSSQL, therefore, TS\_SQL cannot be turned off for BASE DSNs.
  - **TS\_SECURITY Y[ES] | T[RUE] | 1 | N[O] | F[ALSE] | 0** — Specifies whether or not to secure SQL statements before processing them. For example, if a DSN is defined to use TS\_SECURITY NO, Federation Security is bypassed. Therefore, when you connect with the DSN, you will be granted all TSSQL privileges – select, insert, update, delete, references, create table, alter table, and drop table. If, however, a DSN is defined to use TS\_SECURITY YES, the privileges you have been granted will take effect.

If TS\_SECURITY is set to YES, TSSQL will be used. YES is the default value;

- **CREDENTIALS\_SEARCH\_ORDER | CSO "(" *cso-value* [ {"", *cso-value*} ... ] ")"** — Specifies whether to use back end credentials owned by the current user (PERSONAL) or shared among many users (SHARED). The DSN may be configured to search for either in the order specified. If a search is not specified, the default is CSO (PERSONAL).

For example, if your user owns a database login and a DSN is configured with a CSO value of PERSONAL, the DSN will use your login credentials to connect to a database. If, however, you have access to a shared login, a DSN will connect with the login credentials that are associated with the shared login if the DSN is configured with a CSO value of SHARED. Users, logins, and shared logins are created using the Authentication Server. For more information, see the *DataFlux Authentication Server User's Guide*.

```
cso-value ::= PERSONAL | SHARED
```

## Examples

```
CREATE DSN "DSN1" UNDER BASE DESCRIPTION 'creating DSN1' NOPROMPT
'DRIVER=BASE;CATALOG="catalog1_BASE";SCHEMA=(name="schemal_BASE")' {OPTIONS
(TS_SQL NO,TS_SECURITY NO)}

create dsn BASEDSN under BASE NOPROMPT 'DATA_SERVICE=BASE;LOCKTABLE=EXCLUSIVE'
```

```

create dsn BASEDSN under BASE CONNECT
'CATALOG="catalog1_BASE";LOCKTABLE=EXCLUSIVE'

create dsn BASEDSN under BASE NOPROMPT
'(CATALOG="catalog1_BASE";LOCKTABLE=SHARE);(CATALOG="catalog2_BASE";LOCKTABLE=EXCLUSIVE)'

create dsn BASEDSN under BASE CONNECT
'DRIVER=TSSQL;CONOPTS=((CATALOG="catalog1_BASE";LOCKTABLE=SHARE);(CATALOG="catalog2_BASE";LOCKTABLE=EXCLUSIVE))'

create dsn BASEDSN under BASE NOPROMPT
'CATALOG="catalog1_BASE";LOCKTABLE=SHARE;SCHEMA=(NAME="schema1_BASE";LOCKTABLE=EXCLUSIVE)'

create dsn BASEDSN under BASE CONNECT
'CATALOG="catalog1_BASE";LOCKTABLE=SHARE;SCHEMA=(NAME="schema1_BASE";LOCKTABLE=EXCLUSIVE);SCHEMA=(NAME="schema2_BASE";ACCESS=TEMP)'

create dsn SQLSVRDSN_DS1 under SQLServer1 CONNECT
'DATA_SERVICE=SQLSERVER1;STRIP_BLANKS=N'

create dsn MYDSN under MYSERV {OPTIONS CREDENTIALS_SEARCH_ORDER(PERSONAL)}

create dsn MYDSN under MYSERV {OPTIONS CREDENTIALS_SEARCH_ORDER(PERSONAL,SHARED)}

```

## DROP DSN Statement

Allows you to drop server-based DSN.

### Syntax

```
DROP DSN dsn-name [FORCE]
```

### Syntax Description

- ***dsn-name*** — Specifies the DSN name.
- **FORCE** — Specifies the optional FORCE keyword that will suppress error messages when the DSN does not exist.

### Examples

```

DROP DSN "DSN1"

DROP DSN "DSN1" FORCE

```

## ALTER DSN Statement

Allows you to change the server-based DSN. You can change the name and alter advanced options that are driver-specific. For information on which advanced options are supported for each data service, see the data service reference chapter for your driver. Information specific to the data service you are configuring can be found in the section for your data service.

## Syntax

```
ALTER DSN dsn-name alter-dsn-options
ALTER DSN dsn-name RENAME TO new-dsn-name
```

## Syntax Description

- ***dsn-name*** — Specifies the DSN name.
- ***alter-dsn-options*** — Specifies the options to alter the schema.

```
alter-dsn-options
    ::= create-dsdsn-options
```

- ***new-dsn-name*** — Specifies the new DSN name.

## Examples

```
ALTER DSN "DSN1" DESC 'altering DSN1 description' NOPROMPT
'DRIVER=BASE;CATALOG="catalog1_BASE";SCHEMA=(name="schema1_BASE")'

ALTER DSN "DSN5" RENAME to DSN7

ALTER DSN "DSN7" {OPTIONS set (TS_SQL YES,TS_SECURITY YES)}

ALTER DSN "DSN7" {OPTIONS xset CREDENTIALS_SEARCH_ORDER(SHARED), xset TS_SQL
NO, xset TS_SECURITY NO}

ALTER DSN "DSN7" {OPTIONS DROP TS_SQL, DROP TS_SECURITY}
```

## DROP AUTHID Statement

Allows you to drop authorization identifiers.

## Syntax

```
DROP { AUTHID | AUTHORIZATION [IDENTIFIER] } "id" [TRANSFER TO name]
[CASCADE|RESTRICT] [FORCE]
```

## Syntax Description

- ***id*** — Specifies the authorization identity to drop. The ID must be in quotation marks.

This is the user ID or the group ID that you can find by using the Federation Server AUTHORIZATION\_IDENTIFIERS information view. For more information, see [AUTHORIZATION\\_IDENTIFIERS View](#).

- ***name*** — Specifies the authorization identity that will receive object ownership from the dropped identity. This user name is created using the Authentication Server. For more information, see *DataFlux Authentication Server User's Guide*.
- **FORCE** — Specifies the optional FORCE keyword that will suppress error messages when the user does not exist.
- **CASCADE** — Specifies the entities are dropped unconditionally. All records that reference the entity are removed. This option is invalid if the TRANSFER TO option is specified.

- **RESTRICT** — Specifies the drop fails if the entity is the grantor of any privilege. The drop also fails if the entity is the owner of a DSN or schema. This option is ignored if the TRANSFER TO option is specified.

## Examples

```
drop authid "5E563F78B0D70854086FB3D8441EF9AA" transfer to user2

drop authid "F135005B80DED494E996F70DCC53790D" cascade

drop authid "B8A105927F25B1A47AE8198D1E3C4B86" transfer to user2 force
```

## GRANT and DENY Statements

Allows you to give privileges to a specific user or all users to perform actions on objects.

 **Note:** CREATE DSN and ADMINISTER privileges cannot be granted and denied for the PUBLIC and USERS groups.

### Syntax

```
GRANT | DENY { { objectpriv | containerpriv | serverpriv [,...] } |
              ALL [ PRIVILEGES ] }
  [ ON { SCHEMA schemaname | CATALOG catalogname | [DATA] SERVICE servicename
    | DSN dsnname | SERVER } ]
  TO { authid | PUBLIC | USERS } [, ...]
  [ AS ADMINISTRATOR ]
```

### Syntax Description

- ***objectpriv*** — Specifies the name of an object-level privilege to grant or deny, as one of the following values:
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
  - REFERENCES
- ***containerpriv*** — Specifies the name of a container-level privilege to grant or deny, as one of the following values:
  - CREATE TABLE
  - ALTER TABLE
  - DROP TABLE
- ***serverpriv*** — Specifies the name of the server-level privilege to grant or deny, as one of the following values:
  - ADMINISTER
  - TRACE

- CREATE DSN
- CONNECT
- **schemaname** — Specifies the name of the schema.
- **catalogname** — Specifies the name of the catalog.
- **servicename** — Specifies the name of the service.
- **dsnname** — Specifies the name of the DSN.
- **authid** — Specifies the user or group name for which the privileges are granted or denied.
- **AS ADMINISTRATOR** — Grants privileges using the ADMINISTRATOR role. Without this, the privilege is granted as the individual user.



**Note:** If the user is a SYSTEM user, then privileges are assigned with SYSTEM as the grantor.

The following table lists the valid privilege and object combinations, which are indicated by a check mark ( ✓ ):

Privilege	Schema	Catalog	Data Service	DSN	Server
SELECT	✓	✓	✓	✗	✓
INSERT	✓	✓	✓	✗	✓
UPDATE	✓	✓	✓	✗	✓
DELETE	✓	✓	✓	✗	✓
REFERENCES	✓	✓	✓	✗	✓
CREATE TABLE	✓	✓	✓	✗	✓
ALTER TABLE	✓	✓	✓	✗	✓
DROP TABLE	✓	✓	✓	✗	✓
ADMINISTER	✗	✗	✗	✗	✓
TRACE	✗	✗	✗	✗	✓
CREATE DSN	✗	✗	✓	✗	✓
CONNECT	✗	✗	✓	✓	✓

## Examples

```
GRANT INSERT ON SCHEMA "BASE_CATALOG1"."schema1_BASE" TO user1
```

```
grant connect on server to user1
```

```
deny connect on DSN SQLSRVDSN1 to user1
```

```
grant create dsn on data service SQLSRV1 to user1
```

```
grant administer on server to user1
```

```
DENY ALL ON SCHEMA "BASE_CATALOG1"."schema1_BASE" TO user1
```

# REVOKE Statement

Allows you to remove explicitly granted or denied privileges from the specified object.



**Note:** CREATE DSN and ADMINISTER privileges cannot be revoked for the PUBLIC and USERS groups.

## Syntax

```
REVOKE { { objectpriv | containerpriv | serverpriv [,...] } |  
ALL [ PRIVILEGES ] }  
[ ON { SCHEMA schemaname | CATALOG catalogname | [DATA] SERVICE servicename |  
DSN dsnname | SERVER } ]  
FROM { authid | PUBLIC | USERS } [, ...]
```

## Syntax Description

- ***objectpriv*** — Specifies the name of an object-level privilege to grant or deny, as one of the following values:
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
  - REFERENCES
- ***containerpriv*** — Specifies the name of a container-level privilege to grant or deny, as one of the following values:
  - CREATE TABLE
  - ALTER TABLE
  - DROP TABLE
- ***serverpriv*** — Specifies the name of the server-level privilege to grant or deny, as one of the following values:
  - ADMINISTER
  - TRACE
  - CREATE DSN
  - CONNECT
- ***schemaname*** — Specifies the name of the schema.
- ***catalogname*** — Specifies the name of the catalog.
- ***servicename*** — Specifies the name of the service.
- ***dsnname*** — Specifies the name of the DSN.

- **authid** — Specifies the user or group name for which the privileges are granted or denied.
- **AS ADMINISTRATOR** — Grants privileges using the ADMINISTRATOR role. Without this, the privilege is granted as the individual user.



**Note:** If the user is a SYSTEM user, then privileges are assigned with SYSTEM as the grantor.

## Examples

```
REVOKE ALL ON SCHEMA "BASE_CATALOG1"."schemal_BASE" FROM user1
```

```
REVOKE INSERT ON DATA SERVICE BASE FROM USER1
```

```
revoke all on server from user1
```



# Appendix B: Federation Information Views

You can select from the following information views to get information about what is stored in the syscat.tdb database, which is used by the DataFlux Federation Server (Federation Server):

- [Visibility of Information Views](#)
- [AUTHORIZATION IDENTIFIERS View](#)
- [IDENTITY View](#)
- [CATALOG PRIVILEGES View](#)
- [CATALOGS View](#)
- [DATA SERVICES View](#)
- [SCHEMAS View](#)
- [DS PRIVILEGES View](#)
- [SCHEMA PRIVILEGES View](#)
- [DSN PRIVILEGES and EFFECTIVE DSN PRIVILEGES Views](#)
- [DATA SOURCE NAMES View](#)
- [DSN CONTENT View](#)
- [CONFIG DSNS View](#)
- [CONFIG SCHEMAS View](#)
- [CONFIG DATA SERVICES View](#)
- [CONFIG CATALOGS View](#)
- [PRIVILEGES and EFFECTIVE PRIVILEGES Views](#)

## Visibility of Information Views

The following section describes what gets returned from the information views for administrators and users with different privileges:

### Administrators

SYSTEM and Federation Server administrators can view all information, which includes users, groups, and objects.

## Users

All users can query views. However, the privileges the user has will determine what records the user will see. The following list specifies what objects are visible to a user:

- A data source name (DSN) is visible to a user who is the owner of the DSN or has Connect privilege on the DSN.
- Data services in the DATA\_SERVICES and CONFIG\_DATA\_SERVICES information views are available to users that have Create DSN or Connect privilege on those data services. A user that has Connect privilege on, or is an owner of any DSN in the data service can also view the data service.
- A schema is visible to a user if both of the following conditions exist:
  1. The user has Connect, Administer, or Create DSN privileges (\*) on the containing data service
  2. The user has privileges (\*) on the schema
- A catalog is visible to a user if both of the following conditions exist:
  1. The user has Connect, Administer, or Create DSN privileges (\*) on the containing data service
  2. The user has privileges (\*) on the catalog or any of the schemas associated with the catalog

## AUTHORIZATION\_IDENTIFIERS View

Displays the Authentication Server groups, roles, and users that have been created using the Authentication Server even if those groups, roles, and users have since been deleted from the Authentication Server. Removing a group, role, or user from the Authentication Server does not automatically remove it from the AUTHORIZATION\_IDENTIFIERS view. You must use the DROP AUTHID | AUTHORIZATION IDENTIFIER DDL to clean up orphaned groups, roles, and users from the AUTHORIZATION\_IDENTIFIERS view that have been deleted from the Authentication Server. For more information, see [DROP AUTHID Statement](#).

The following table lists the columns that will be displayed:

Name	Type	Description
name	varchar(256)	Specifies the name of a group, role, or user. Users and groups are created using the Authentication Server. NULL is displayed if the group, role, or user is orphaned.
id	varchar(256)	Specifies the authorization identifier from the Authentication Server.
type	char(1)	Specifies the type of entity as one of the following values: <ul style="list-style-type: none"><li>• G — Indicates the entity is a group.</li><li>• R — Indicates the entity is a role.</li><li>• U — Indicates the entity is a user.</li></ul>

## IDENTITY View

Displays identity information about the connected Federation Server connected user.

The following table lists the columns that will be displayed:

Name	Type	Description
user_name	varchar(256)	Specifies the name of the user that was created with the Authentication Server.
auth_domain	varchar(256)	Specifies the domain name of the authenticated user. For example, if you connect to the Federation Server with the local\myuser account, the auth_domain is <i>local</i> .
auth_id	varchar(256)	Specifies the user name for the authenticated user. For example, if you connect to the Federation Server with the local\myuser account, the auth_id is <i>myuser</i> .

## CATALOG\_PRIVILEGES View

Displays the catalog-level privileges for each catalog.

The following table lists the columns that will be displayed:

Name	Type	Description
catalog_name	varchar(256)	Specifies the name of the catalog.
grantor	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantor_type	char(1)	Specifies the grantor type as one of the following values: <ul style="list-style-type: none"><li>• R — Indicates the grantor is a role.</li><li>• U — Indicates the grantor is a user.</li></ul>
grantee	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantee_type	char(1)	Specifies the grantee type as one of the following values: <ul style="list-style-type: none"><li>• G — Indicates the grantee is a group.</li><li>• U — Indicates the grantee is a user.</li></ul>
privilege_name	varchar(20)	Specifies the privilege name as one of the following values: <ul style="list-style-type: none"><li>• SELECT</li><li>• UPDATE</li></ul>

Name	Type	Description
		<ul style="list-style-type: none"> <li>• INSERT</li> <li>• DELETE</li> <li>• REFERENCES</li> <li>• CREATE TABLE</li> <li>• ALTER TABLE</li> <li>• DROP TABLE</li> </ul>
privilege_type	varchar(5)	Specifies the privilege type as <b>GRANT</b> or <b>DENY</b> .
grantable	char(1)	Specifies whether or not the user can grant this privilege to other users. The only valid value is <b>N</b> .

## CATALOGS View

Displays information about the name and system identifier for each catalog.

The following table lists the columns that will be displayed:

Name	Type	Description
catalog_name	varchar(256)	Specifies the name of the catalog.
data_service_name	varchar(256)	Specifies the name of the data service to which the catalog is associated.
native_catalog_name	varchar(256)	Specifies the name of the native catalog. If not applicable, the value is NULL.

## DATA\_SERVICES View

Displays information about each data service. It also shows a single entry for the Federation Server. For the Federation Server itself, the value of the *data\_service\_name* column will be *\_SERVER\_* and the value of the *type* column will be *SERVER*.

The following table lists the columns that will be displayed:

Name	Type	Description
data_service_name	varchar(256)	Specifies the unique name of the data service.
version	char(32)	Specifies the version of the data source.
type	char(32)	Specifies the data service type. Valid values are: <ul style="list-style-type: none"> <li>• BASE</li> </ul>

Name	Type	Description
		<ul style="list-style-type: none"> <li>• ORACLE</li> <li>• DB2UNXPC</li> <li>• SQLSVR</li> <li>• MYSQL</li> <li>• TERADATA</li> </ul>
domain	varchar(256)	Specifies the domain that is associated with the data service. A domain name is specified when creating a data service. The domain is created using the Authentication Server. In the Authentication Server, any logins that are associated with this domain can potentially be used to connect to the data service using the Federation Server.

## SCHEMAS View

Displays information about each schema.

The following table lists the columns that will be displayed:

Name	Type	Description
schema_name	varchar(256)	Specifies the name of the schema.
catalog_name	varchar(256)	Specifies the name of the catalog. If not applicable, the value is NULL.
user_name	varchar(256)	Specifies NULL.

## DS\_PRIVILEGES View

Displays the data service level privileges for each data service.

The following table lists the columns that will be displayed:

Name	Type	Description
data_service_name	varchar(256)	Specifies the unique data service name.
grantor	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantor_type	char(1)	Specifies the grantor type as one of the following values: <ul style="list-style-type: none"> <li>• R — Indicates the grantor is a role.</li> <li>• U — Indicates the grantor is a user.</li> </ul>

Name	Type	Description
grantee	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantee_type	char(1)	Specifies the grantee type as one of the following values: <ul style="list-style-type: none"> <li>G — Indicates the grantee is a group.</li> <li>U — Indicates the grantee is a user.</li> </ul>
privilege_name	varchar(20)	Specifies the privilege name as one of the following values: <ul style="list-style-type: none"> <li>SELECT</li> <li>UPDATE</li> <li>INSERT</li> <li>DELETE</li> <li>REFERENCES</li> <li>CREATE TABLE</li> <li>ALTER TABLE</li> <li>DROP TABLE</li> <li>CREATE DSN</li> <li>ACCOUNT REGISTRATION</li> <li>CONNECT</li> </ul> <p>If the value of data_service_name is <i>_SERVER_</i>, which corresponds to the Federation Server, the ADMINISTER and TRACE privileges can also be displayed. The ADMINISTER and TRACE privileges can only be set on the Federation Server.</p>
privilege_type	varchar(5)	Specifies the privilege type as <b>GRANT</b> or <b>DENY</b> .
grantable	char(1)	Specifies whether or not the user can grant this privilege to other users. The only valid value is <b>N</b> .

## SCHEMA\_PRIVILEGES View

Displays the schema-level privileges for each schema.

The following table lists the columns that will be displayed:


Name	Type	Description
catalog_name	varchar(256)	Specifies the name of the catalog. If not applicable, the value is NULL.
schema_name	varchar(256)	Specifies the name of the schema.
grantor	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantor_type	char(1)	Specifies the grantor type as one of the following values: <ul style="list-style-type: none"> <li>• R — Indicates the grantor is a role.</li> <li>• U — Indicates the grantor is a user.</li> </ul>
grantee	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantee_type	char(1)	Specifies the grantee type as one of the following values: <ul style="list-style-type: none"> <li>• G — Indicates the grantee is a group.</li> <li>• U — Indicates the grantee is a user.</li> </ul>
privilege_name	varchar(20)	Specifies the privilege name as one of the following values: <ul style="list-style-type: none"> <li>• SELECT</li> <li>• UPDATE</li> <li>• INSERT</li> <li>• DELETE</li> <li>• REFERENCES</li> <li>• CREATE TABLE</li> <li>• ALTER TABLE</li> <li>• DROP TABLE</li> </ul>
privilege_type	varchar(5)	Specifies the privilege type as <b>GRANT</b> or <b>DENY</b> .
grantable	char(1)	Specifies whether or not the user can grant this privilege to other users. The only valid value is <b>N</b> .

## DSN\_PRIVILEGES and EFFECTIVE\_DSN\_PRIVILEGES Views

Displays privileges for users and groups on each data source name (DSN) and indicates inheritance. Both views show all direct (explicit) and inherited privileges based on the privileges of the user and group, or its group membership.

The DSN\_PRIVILEGES result set contains rows for users and groups that have the connect privilege explicitly set on either the server, service, or DSN. If a user or group does not have any direct privilege, it will not be shown in this view. It is a condensed view of the EFFECTIVE\_DSN\_PRIVILEGES view.

The EFFECTIVE\_DSN\_PRIVILEGES result set contains rows for all users and groups that have any privilege directly set or a privilege can be derived from its group membership. For example, if a user does not have any privileges set on any of the Federation Server objects, the user will still be in the result set if the user is a member of a group that has a direct privilege set.

 **Note:** Both of these views can return very large result sets depending on the configuration of Federation Server. Subsetting on DATA\_SERVICE, CATALOG\_NAME, and/or SCHEMA\_NAME can reduce the size of the result set.

The following table lists the columns that will be displayed:

Name	Type	Description
dsn_name	varchar(256)	Specifies the unique name of the DSN.
data_service	varchar(256)	Specifies the name of the data service.
grantor_id	varchar(256)	Specifies the ID of the user that granted or denied the privilege.
grantor	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantor_type	char(1)	Specifies the grantor type as one of the following values: <ul style="list-style-type: none"> <li>• R — Indicates the grantor is a role.</li> <li>• U — Indicates the grantor is a user.</li> </ul>
grantee_id	varchar(256)	Specifies the ID of the user that is granted or denied the privilege.
grantee	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantee_type	char(1)	Specifies the grantee type as one of the following values: <ul style="list-style-type: none"> <li>• G — Indicates the grantee is a group.</li> <li>• U — Indicates the grantee is a user.</li> </ul>
privilege_name	varchar(20)	Specifies the CONNECT privilege name.
privilege_type	varchar(5)	Specifies the privilege type as <b>GRANT</b> or <b>DENY</b> .
grantable	char(1)	Specifies whether or not the user can grant this privilege to other users. The only valid value is <b>N</b> .



Name	Type	Description
inherited	char(1)	Specifies whether or not the privilege is inherited, as either <b>Y</b> or <b>N</b> .
source_object_level	integer	Specifies the object level where the privilege is inherited, as one of the following values: <ul style="list-style-type: none"> <li>0 — server</li> <li>1 — data service</li> <li>2 — DSN</li> </ul>
source_grantee_id	varchar(256)	Specifies the ID of a group or user from which the privilege is derived.
source_grantee	varchar(256)	Specifies the name of the group or user from which the privilege is derived.
source_grantee_type	char(1)	Specifies the source_grantee type as one of the following values: <ul style="list-style-type: none"> <li>G — Indicates the source_grantee is a group.</li> <li>U — Indicates the source_grantee is a user.</li> </ul>

## DATA\_SOURCE\_NAMES View

Displays one entry per configured DSN.

The following table lists the columns that will be displayed:

Name	Type	Description
dsn_name	varchar(256)	Specifies the unique name of the DSN.
data_service_name	varchar(256)	Specifies the unique name of the data service.
Desc	varchar(256)	Specifies the descriptive text.
Format	char(32)	Specifies the format of the content as <i>STANDARD</i> , which is the standard driver connection string.
owner_name	varchar(256)	Specifies the name of the user that owns the DSN.
owner_id	varchar(256)	Specifies the ID of the user that owns the DSN.
owner_type	char(1)	Specifies the owner type as one of the following values: <ul style="list-style-type: none"> <li>R — Indicates the owner is a role.</li> <li>U — Indicates the owner is a user.</li> </ul>
dsn_type	varchar(256)	Specifies the DSN type as either <i>CONNECT</i> or <i>NOPROMPT</i> .

## DSN\_CONTENT View

Displays one or more rows per configured DSN. Each row contains a portion of the DSN content, ordered by sequence. If `DATA_SOURCE_NAMES.format` is *STANDARD*, then the content column contains driver connection string syntax.

The following table lists the columns that will be displayed:

Name	Type	Description
dsn_name	varchar(256)	Specifies the unique name of the DSN.
sequence	integer	Specifies the configuration chunk sequence.
content	varchar(1024)	Specifies the DSN content.

## CONFIG\_DSNS View

Displays generic configuration variables for each DSN. All configuration settings for a single DSN may be obtained by concatenating rows that match the correct `dsn_name` and ordering the results by sequence.

The following table lists the columns that will be displayed:

Name	Type	Description
data	varchar(128)	Specifies the configuration data. The value cannot be NULL.
dsn_name	varchar(256)	Specifies the unique name of the DSN.
sequence	smallint	Specifies the configuration chunk sequence.

## CONFIG\_SCHEMAS View

Displays generic configuration variables for each schema. All configuration settings for a single schema may be obtained by concatenating rows matching the correct `catalog_name` and `schema_name` and ordering the results by sequence.

The following table lists the columns that will be displayed:

Name	Type	Description
data	varchar(128)	Specifies the configuration data. The value cannot be NULL.
catalog_name	varchar(256)	Specifies the name of the catalog. If not applicable, the value is NULL.
schema_name	varchar(256)	Specifies the name of the schema.
sequence	smallint	Specifies the configuration chunk sequence.

## CONFIG\_DATA\_SERVICES View

Displays generic configuration variables for each data service. All configuration settings for a single service may be obtained by concatenating rows matching the correct `data_service_name` and ordering the results by sequence.

The following table lists the columns that will be displayed:

Name	Type	Description
data	varchar(128)	Specifies the configuration data. The value cannot be NULL.
data_service_name	varchar(256)	Specifies the unique name of the data service.
sequence	smallint	Specifies the configuration chunk sequence.

## CONFIG\_CATALOGS View

Displays generic configuration variables for each catalog. All configuration settings for a single catalog may be obtained by concatenating values in the data column for every row with the same `catalog_name`.

The following table lists the columns that will be displayed:

Name	Type	Description
data	varchar(128)	Specifies the configuration data as not NULL.
catalog_name	varchar(256)	Specifies the name of the catalog.
sequence	smallint	Specifies the configuration chunk sequence.


## PRIVILEGES and EFFECTIVE\_PRIVILEGES Views

Displays the privileges for users and groups on all objects, and indicates inheritance. The objects include Federation Server, data services, catalogs, and schemas. Both views show all direct (explicit) and inherited privileges based on the privileges of the user and group, or its group membership.

The `PRIVILEGES` result set contains rows for users and groups that have any privilege directly set. If a user or group does not have any direct privilege, it will not be shown in this view. It is a condensed view of the `EFFECTIVE_PRIVILEGES` view.

The `EFFECTIVE_PRIVILEGES` result set contains rows for all users and groups that have any privilege directly set or a privilege can be derived from its group membership. For example, if a user does not have any privileges set on any of the Federation Server objects, the user will still be in the result set if the user is a member of a group that has a direct privilege set.

By default, if a privilege is not explicitly listed in the result sets, it is denied.

 **Note:** Both of these views can return very large result sets depending on the configuration of Federation Server. Subsetting on DATA\_SERVICE, CATALOG\_NAME, and/or SCHEMA\_NAME can reduce the size of the result set.

The following table lists the columns that will be displayed:

Name	Type	Description
data_service	varchar(256)	Specifies the name of the data service.
catalog_name	varchar(256)	Specifies the name of the catalog.
schema_name	varchar(256)	Specifies the name of the schema.
grantor_id	varchar(256)	Specifies the ID of the user that granted or denied the privilege.
grantor	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantor_type	char(1)	Specifies the grantor type as one of the following values: <ul style="list-style-type: none"> <li>• R — Indicates the grantor is a role.</li> <li>• U — Indicates the grantor is a user.</li> </ul>
grantee_id	varchar(256)	Specifies the ID of the user that is granted or denied the privilege.
grantee	varchar(256)	Specifies the name of the user who is granted or denied the privilege.
grantee_type	char(1)	Specifies the grantee type as one of the following values: <ul style="list-style-type: none"> <li>• G — Indicates the grantee is a group.</li> <li>• U — Indicates the grantee is a user.</li> </ul>
privilege_name	varchar(20)	Specifies the privilege name as one of the following values: <ul style="list-style-type: none"> <li>• SELECT</li> <li>• UPDATE</li> <li>• INSERT</li> <li>• DELETE</li> <li>• REFERENCES</li> <li>• CREATE TABLE</li> <li>• ALTER TABLE</li> <li>• DROP TABLE</li> <li>• ADMINISTER</li> </ul>

Name	Type	Description
		<ul style="list-style-type: none"> <li>• TRACE</li> <li>• CREATE DSN</li> <li>• ACCOUNT REGISTRATION</li> <li>• CONNECT</li> </ul>
privilege_type	varchar(5)	Specifies the privilege type as <b>GRANT</b> or <b>DENY</b> .
grantable	char(1)	Specifies whether or not the user can grant this privilege to other users. The only valid value is <b>N</b> .
inherited	char(1)	Specifies whether or not the privilege is inherited, as either <b>Y</b> or <b>N</b> .
source_object_level	integer	<p>Specifies the object level where the privilege is inherited, as one of the following values:</p> <ul style="list-style-type: none"> <li>• 0 — server</li> <li>• 1 — data service</li> <li>• 2 — catalog</li> <li>• 3 — schema</li> </ul>
source_grantee_id	varchar(256)	Specifies the ID of a group or user from which the privilege is derived.
source_grantee	varchar(256)	Specifies the name of the group or user from which the privilege is derived.
source_grantee_type	char(1)	<p>Specifies the source_grantee type as one of the following values:</p> <ul style="list-style-type: none"> <li>• G — Indicates the source_grantee is a group.</li> <li>• U — Indicates the source_grantee is a user.</li> </ul>

# Appendix C: Code Example

The following example shows you how to create a materialized view.

## SAS\_CACHED\_CATALOG File

```
/*-----SAS_CACHED_CATALOG.SQL-----*/
/* This script is used to create the materialized and the synonym needed to
   get the ForeignKey metadata.

   Please work with your DBA to set this up. Materialized views can be complex
   and so thorough understanding will help us use them effectively. Especially
   deciding how to do the refreshes. Here we provide the simplest possible
   steps
   to create the required materialized view and the command to refresh it
   manually.

   The materialized view below can be created in any schema with any name.
   Feel free to add whatever REFRESH options suits your purpose. Note that you
   may
   need additional steps based on the REFRESH option setting. Here we provide
   the
   simplest possible way to do this.

   The PUBLIC synonym pointing to this Materialized view must be named
   "SAS_CACHED_FK_CATALOG_PSYN". This synonym must be visible to PUBLIC(or the
   set
   of users who will be needing ForeignKey metadata) so that it is
   accessible from any
   schema.
*/

Create materialized view SAS_CACHED_FK_CATALOG_MATVIEW REFRESH ON DEMAND
as SELECT
    PKAC.OWNER as PKTABLE_SCHEM,
    PKAC.TABLE_NAME as PKTABLE_NAME,
    PKACC.COLUMN_NAME as PKCOLUMN_NAME,
    FKAC.OWNER as FKTABLE_SCHEM,
    FKAC.TABLE_NAME as FKTABLE_NAME,
    FKACC.COLUMN_NAME as FKCOLUMN_NAME,
    FKACC.POSITION as KEY_SEQ,
    FKAC.CONSTRAINT_NAME as FK_NAME,
    PKAC.CONSTRAINT_NAME as PK_NAME
from
    sys.all_constraints PKAC, sys.all_constraints FKAC,
    sys.all_cons_columns PKACC, sys.all_cons_columns FKACC
where
    FKAC.r_constraint_name=PKAC.constraint_name and
    FKAC.constraint_name=FKACC.constraint_name and
    PKAC.constraint_name=PKACC.constraint_name and PKAC.constraint_type='P' and
    FKAC.constraint_type='R' and FKAC.owner=FKACC.owner and PKAC.owner=PKACC.owner
    and PKAC.table_name=PKACC.table_name and FKAC.table_name=FKACC.table_name and
    FKACC.position = PKACC.position ;

/* The synonym name *must* be SAS_CACHED_FK_CATALOG_PUBLIC_SYNONYM */
create public synonym SAS_CACHED_FK_CATALOG_PSYN for
SAS_CACHED_FK_CATALOG_MATVIEW;
```

```
grant all on SAS_CACHED_FK_CATALOG_PSYN to PUBLIC;

/*-----Manual REFRESH of the Materialized View-----
*/
/* Note there are several ways to do this, consult with your DBA.
   Here are a couple of ways:
*/
execute DBMS_MVIEW.REFRESH('SAS_CACHED_FK_CATALOG_MATVIEW');
execute DBMS_SNAPSHOT.REFRESH('SAS_CACHED_FK_CATALOG_MATVIEW', '?');
```

# Glossary

## A

---

### **ACID**

Acronym for Atomicity, Consistency, Isolation, Durability: features of a Transactional Data Store

### **advanced options**

Advanced options are additional options that are not required for establishing a connection to a data service. While advanced options can be specified when creating a data service, in most cases, they are specified for catalogs, data source names, and schemas.

### **ANSI**

The American National Standards Institute (ANSI) is an organization in the United States that coordinates voluntary standards and conformity to those standards.

### **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.

### **ARM**

Application Response Measurement (ARM) is an application programming interface that was developed by an industry partnership and which is used to monitor the availability and performance of software applications. ARM monitors the application tasks that are important to a particular business.

### **authentication**

The process of verifying the identity of a person or process within the guidelines of a specific security policy.

### **Authentication Data Store**

A component of the Data Management Platform that provides a central location for the management of connections between the Data Management Studio client, the DataFlux Federation and Data Management Servers, and native database servers.

### **authorization**

The process of determining which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.

## C

---

### **client**

A computer or application that requests services, data, or other resources from a server.

### **connection options**

Connection options are used to establish a connection to a data service. One or more connection options must be specified when creating a data service, in order for a connection to be established to the data service. Conversely, connection options should not be specified when creating catalogs, data source names, and schemas.

### **connection string**

The information that defines how to connect an application to the data. In the DataFlux Federation Server, a connection string identifies the information that is required to connect to one or more data services.



## D

---

### **data service**

Contain information about how to connect to a particular source of data, such as Oracle or Base SAS data sets. Data services form the foundation for connectivity to a source of data, and privileges may be assigned to data services to control which users may connect to the given data service.

### **data source**

The generic term for where data resides (the location of the data), relative to data access. Data source refers to the data that the user wants to access and its associated data source driver that parses and executes SQL statements and accesses the physical data.

### **data type**

An attribute of every column in a table that tells the data service how much physical space to set aside for the column, and that specifies what type of data the column contains.

### **DBMS**

Database Management System; controls the creation, maintenance and use of a database.

### **driver**

A special-purpose software program that enables two disparate software programs such as an application and an API to interact.

### **DSN**

A data source name (DSN) is a symbolic name that encapsulates the information that is contained in a connection string. The data source name contains the Federation Server driver name, the physical location of the data, authentication information that is required to retrieve data, such as the credentials for each user.

## E

---

### **encryption**

The act or process of converting data to a form that only the intended recipient can read or use.

## F

---

### **federated connection**

One connection that can connect to two different data services.

### **Federation Server driver**

Software that interacts with a data service in order to read from and write to proprietary file formats.

## I

---

### **IOM**

An Integrated Object Model (IOM) is a set of object-based interfaces to features or services that are provided by Base SAS software. IOM enables application developers to use industry-standard programming languages, programming tools, and communication protocols to develop client programs that access these services on IOM servers.

## J

---

### **join**

To combine data from two or more tables into a single result table.

## L

---

### **locking**

The process by which a data service restricts access to a file or a record in a multi-user environment. The first user who accesses the file or record prevents other users from accessing it.

## 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

---

### **orphaned**

User or group accounts that have been deleted on the Authentication Server but still exist on the Federation Server. Run the DROP AUTHID statement to delete the user or group accounts on the Federation Server.

## R

---

### **result set**

The set of rows or records that a server or other application returns in response to a query.

## S

---

### **scrollable cursor**

A device that enables an application to position on any row in a result set. For example, a scrollable cursor can back up and revisit a row, start at the end of the file and work backward, skip some rows, or go directly to a specific row.

### **SQL**

Structured Query Language (SQL) is a language used to request information from database systems. The DataFlux Federation Server implements SQL through the Federation Server TSSQL language (TSSQL), which can be used to query data services.

## T

---

### **thread**

A single path of execution of a process in a single CPU, or a basic unit of program execution in a thread-enabled operating system. In an SMP environment, which uses multiple CPUs, multiple threads can be spawned and processed simultaneously. Regardless of whether there is one CPU or many, each thread is an independent flow of control that is scheduled by the operating system.

### **Transactional Data Store**

A Transactional Data Store is a place to store transactional data, where ACID (Atomicity, Consistency, Isolation, Durability) features are required. The Federation (and Authentication) Server requires a transactional data store to persist metadata as part of its system tables, and to ensure a consistent state of this metadata.

**TSSQL**

Federation Server TSSQL (TSSQL) is the implementation of SQL used by threaded table services. The TSSQL language is a subset of the ANSI SQL:1999 standard. TSSQL provides an SQL dialect that accesses data from various data services.