

Version 9



Migrating to Federation Version 9



Migrating to Federation Version 9

Note

Before using this information and the product that it supports, be sure to read the general information under “Notices and trademarks” on page 83.

Contents

Chapter 1. Planning to migrate to WebSphere Federation Server 1

Overview for migrating to WebSphere Federation Server	1
Supported data sources	2
Migrated objects	6
Migration restrictions for federated systems	7
Preparing to migrate to federated systems	15
Copying configuration settings	15
Recording database information before migration	16
Recording instance information before migration	17

Chapter 2. Migrating Windows servers 19

Migrating to WebSphere Federation Server on Windows	19
---	----

Chapter 3. Migrating Linux and UNIX servers 23

Migrating to WebSphere Federation Server on Linux and UNIX	23
Migrating 32-bit federated systems to 64-bit federated systems on UNIX	26

Chapter 4. Accessing data sources after migration 29

Configuring federated systems after migration.	29
Restoring configuration settings for federated systems.	29
Converting type-1 indexes to type-2 indexes	30
Rebinding packages	30
Changing the wrapper for a data source.	31
Setting isolation-level behavior for applications after migration	33
Configuring XML data sources after migration	33
Restoring connections to DB2 family data sources	34
Allowing write operations on large object data types after migrating to WebSphere Federation Server	34
Accessing DB2 family data sources after migrating to WebSphere Federation Server	36
Accessing Informix data sources after migrating to WebSphere Federation Server	38
Accessing Microsoft SQL Server data sources after migrating to WebSphere Federation Server	40

Accessing ODBC data sources after migrating to WebSphere Federation Server	41
Accessing Oracle data sources after migrating to WebSphere Federation Server	44
Accessing Sybase data sources after migrating to WebSphere Federation Server	46
Accessing Teradata data sources after migrating to WebSphere Federation Server	47
Modifying existing Teradata configurations.	49
Recreating Teradata configurations	50

Chapter 5. Migration reference 51

DataJoiner migration	51
DataJoiner terms and their federation server equivalents	51
DataJoiner catalog conversion	52
DB2 data sources	53
Oracle data sources.	58
Informix data sources	60
Microsoft SQL data sources	61
ODBC data sources.	63
Sybase data sources	65
Teradata data sources	66
Nonrelational data sources for life sciences	68
Changes to fixed columns for BLAST nicknames	68
Changes to KEGG function arguments	69
Changes to the SSDB function names and accepted arguments	72
Changes to SSDB database queries	73
IMPORT and EXPORT commands for nicknames.	76

Chapter 6. Migrating a replication environment 79

Accessing information about IBM 81

Contacting IBM	81
Accessible documentation	82
Providing comments on the documentation.	82

Notices and trademarks 83

Notices	83
Trademarks	85

Index 87

Chapter 1. Planning to migrate to WebSphere Federation Server

You must plan for the migration by reading about supported data sources and migrated database objects and learning about restrictions and changes that apply when you migrate from DataJoiner[®] or WebSphere[®] Information Integrator.

To migrate to WebSphere Federation Server, version 9.1, you must complete the following tasks:

1. Prepare for migration by backing up databases, recording system and database information, verifying that your databases are ready to migrate, and more.
2. Install required software.
3. Migrate objects such as instances, databases, explain tables, and applications.
4. Rebind packages.
5. Set up access to the data sources.

Overview for migrating to WebSphere Federation Server

You can migrate existing instances and databases to IBM[®] WebSphere Federation Server, version 9.1 from supported versions of DB2[®] DataJoiner, DB2 database systems, and WebSphere Information Integrator, version 8. However, direct migration to WebSphere Federation Server, version 9.1 is supported only from WebSphere Information Integrator, version 8.2.

Supported migration paths

After you install WebSphere Federation Server, you must migrate all instances and databases that you want to access from WebSphere Federation Server.

Migration support for WebSphere Federation Server and DB2 products

You can migrate to WebSphere Federation Server, version 9.1 from the following products:

- DB2 DataJoiner, version 2.1.1 for AIX[®], Solaris, and Windows
If you are using DB2 DataJoiner, version 1.2 or version 2.1, you must upgrade to DB2 DataJoiner, version 2.1.1, before you can migrate to WebSphere Federation Server.
- DB2 Universal Database[™] for Linux[®], UNIX[®], and Windows[®], version 7 federated systems
Migration support includes federated systems that use DB2 Relational Connect or DB2 Life Sciences Data Connect.
- DB2 Universal Database for Linux, UNIX, and Windows, version 8.2 federated systems
- WebSphere Information Integrator, version 8.2

The currently installed product determines the required steps to migrate your instances and databases to WebSphere Federation Server.

DataJoiner, version 2.1.1 and DB2 Universal Database, version 7 federated systems

Direct migration to WebSphere Federation Server, version 9.1 is not supported. If you have DataJoiner, version 2.1.1 or DB2 Universal

Database, version 7 installed, you must migrate to WebSphere Information Integrator, version 8.2 with the latest fix pack, before you migrate to WebSphere Federation Server.

DB2 Universal Database, version 8 federated systems and WebSphere Information Integrator, version 8.2

You can migrate directly to WebSphere Federation Server, version 9.1, if you already installed WebSphere Information Integrator, version 8.2 installed.

Data sources that do not require migration

If you want to access only DB2 family data sources, Informix® data sources, or OLE DB data sources, you do not need to install and migrate to WebSphere Federation Server. Access to these data sources is provided when you install or migrate to DB2 Version 9.1 for Linux, UNIX, and Windows.

Migration methods

You can migrate instances and databases to WebSphere Federation Server by using migration commands, such as db2ckmig, db2imigr, and db2 migrate database.

Related concepts

“Preparing to migrate to federated systems” on page 15

Before you migrate to IBM WebSphere Federation Server, you must perform tasks such as backing up your databases, recording system configurations, recording database and instance information, changing the diagnostic error level, and verifying that your databases are ready for migration.

Related tasks

“Migrating to WebSphere Federation Server on Linux and UNIX” on page 23

You can migrate directly from IBM WebSphere Information Integrator, version 8.2 on a 32-bit or 64-bit system to WebSphere Federation Server. For AIX and Solaris, you can migrate databases and instances to WebSphere Federation Server on 64-bit systems only. For Linux, you can migrate databases and instances to WebSphere Federation Server on 32-bit and 64-bit systems.

“Migrating to WebSphere Federation Server on Windows” on page 19

You can migrate directly from IBM WebSphere Information Integrator, version 8.2 on a 32-bit or 64-bit system to WebSphere Federation Server. For Windows, you can migrate databases and instances to WebSphere Federation Server on 32-bit and 64-bit systems.

Supported data sources

There are many data sources that you can access using a federated system.

The following table lists the supported data sources:

Table 1. Supported data source versions and access methods.

Data source	Supported versions	Access methods and requirements
BLAST	2.2.3 through 2.2.8 fixpacks supported	BLAST daemon (supplied with the wrapper)
BioRS	5.2.x.x	HTTP
DB2 Database for Linux, UNIX, and Windows	8.1, 8.2, 9.1	DRDA®

Table 1. Supported data source versions and access methods. (continued)

Data source	Supported versions	Access methods and requirements
DB2 Universal Database for z/OS [®]	6.1, 7.1 with the following APARs applied: <ul style="list-style-type: none"> • PQ62695 • PQ55393 • PQ56616 • PQ54605 • PQ46183 • PQ62139 	DRDA
DB2 Universal Database for iSeries [™]	5.2 with the following APARs and PTFs applied: <ul style="list-style-type: none"> • APAR SE06003, PTF SI04582 • APAR SE07533, PTF SI05991 • APAR SE08416, PTF SI07135 • APAR II13348, PTFs SF99502, SI11626, SI11378 	DRDA
DB2 Server for VM and VSE	7.1 (or later) with fixes for APARs for schema functions applied.	DRDA
Entrez	Supported	HTTP.
GenBank	Supported	HTTP. Connection to the NCBI through the Web. Use the Entrez wrapper to access this data source.
HMMER	2.2g, 2.3	HMMER daemon (supplied with the wrapper)

Table 1. Supported data source versions and access methods. (continued)

Data source	Supported versions	Access methods and requirements
Informix	7.31, 8.4, 8.5, 9.4, 10.0	<p>Informix Client SDK V2.81 (or later)</p> <p>On Solaris, the Informix client SDK version 2.81.FC2 is not supported. If you are using the Informix client version 2.81.FC2, update the client to version 2.81.FC2R1 or later.</p> <p>On Windows, the Informix client SDK version 2.81.TC2 or later.</p> <p>On the 64-bit mode zLinux operating system, the Informix client version 2.81.FC1 or 2.81.FC2 is not supported. If you are using one of those client versions, update the client to version 2.81.FC3 or later.</p>
KEGG	KEGG API 3.2	User-defined functions for KEGG
Microsoft® Excel	97, 2000, 2002, 2003	Excel 97, 2000, 2002, or 2003 installed on the federated server
Microsoft SQL Server	2000 SP3 and later service packs on that release, 2005	<p>On Windows, the Microsoft SQL Server Client ODBC 3.0 (or later) driver.</p> <p>On UNIX:</p> <ul style="list-style-type: none"> • DataDirect Technologies (formerly MERANT) Connect ODBC 4.2 (or later) driver. • Microsoft SQL Server wrapper with a UTF-8 database requires DataDirect Connect for ODBC 4.2 Service Pack 2 or later.
ODBC	3.x	ODBC driver for the data source. ODBC driver access to Redbrick and ODBC driver access to WebSphere Classic Federation Server for z/OS data sources, such as IMS™, VSAM, CA-Datacom, CA-IDMS, and Software AG Adabas.
OLE DB	2.7, 2.8	OLE DB 2.0 (or later)

Table 1. Supported data source versions and access methods. (continued)

Data source	Supported versions	Access methods and requirements
OMIM	Supported	HTTP. Connection to the NCBI through the Web and the OMIM query.fcgi utility. Use the Entrez wrapper to access this data source.
Oracle	8.1.7, 9.0, 9.1, 9.2, 9i, 10g	Oracle net client or NET8 client software
PeopleSoft	8.x	IBM WebSphere Business Integration Adapter for PeopleSoft v2.3.1, 2.4. Requires WebSphere MQ Series.
PubMed	Supported	HTTP. Connection to the NCBI through the Web. Use the Entrez wrapper to access this data source.
SAP	3.x, 4.x	IBM WebSphere Business Integration Adapter for mySAP.com v2.3.1, 2.4. Requires WebSphere MQ Series.
Script		Script daemon (supplied with the wrapper)
Siebel	7, 7.5, 2000	IBM WebSphere Business Integration Adapter for Siebel eBusiness Applications v2.3.1, 2.4. Requires WebSphere MQ Series.
Sybase	12.0, 12.5	Sybase Open Client ctlib interface
Table-structured files		None
Teradata	V2R4, V2R5, V2R6	Teradata Call-Level Interface, Version 2 (CLIV2) Release 04.06 (or later) On Windows, the Teradata client TTU 7.0 or later and the Teradata API library CLIV2 4.7.0 or later on the federated server.
Web services	SOAP 1.0., 1.1, WSDL 1.0, 1.1 specifications	HTTP, HTTPS. SOAP user-defined functions consume Web services.
WebSphere MQ	Server edition 6.0	WebSphere MQ user-defined functions in schemas DB2MQ, DB2MQ1C, and DB2MQT.
XML	1.0 specification	None

Migrated objects

When instances and databases are migrated to IBM WebSphere Federation Server, objects are migrated. For objects that are not migrated, you might need to perform tasks after you migrate to WebSphere Federation Server or certain database operations might be limited.

When instances and databases are migrated to WebSphere Federation Server, some objects are updated so that they perform properly in a federated system. These changes include updates to client libraries and file paths.

When instances and databases are migrated to WebSphere Federation Server, the following objects are migrated:

- Wrapper definitions
- Server definitions
- Nicknames

Exception: Stored procedure nicknames are not migrated. After you migrate to WebSphere Federation Server, you can recreate stored procedure nicknames as federated stored procedures by using the CREATE PROCEDURE statement.

- Tables
- User mappings
- Views

Exception: Local views that were created from a nickname by using the WITH CHECK OPTION clause will be inoperative after migration. This syntax is not supported.

- Schema names
- Column names
- Index names
- Data type mappings

Exception: If a data type is not supported in WebSphere Federation Server, the data type is mapped to a supported data type.

- Package names
- User-defined functions
- Function mappings

Exception: When migrating from DataJoiner, function mappings that map a DB2 SYSIBM or SYSFUN function to a remote function are not migrated.

WebSphere Federation Server relational wrappers use a set of default function mappings to map DB2 SYSIBM and SYSFUN functions to remote functions.

Related concepts

Wrappers and wrapper modules

Server definitions and server options

Federated stored procedures

Related tasks

Creating federated stored procedures

Related reference

“DataJoiner terms and their federation server equivalents” on page 51 WebSphere Federation Server uses terms that are different but equivalent to common DataJoiner terms.

Migration restrictions for federated systems

When you migrate to IBM WebSphere Federation Server from other federated systems, certain migration restrictions and changes might apply. You might need to do tasks before or after you migrate to address the restrictions and changes, or you might have to limit certain database operations after migration.

DataJoiner migration restrictions

Before you migrate from DataJoiner to IBM WebSphere Federation Server, there are migration restrictions and changes. You might have to perform tasks before or after you migrate, or you might have to limit certain database operations after migration.

Abstract data types and extended indexes

WebSphere Federation Server does not support databases that contain abstract data types or extended indexes. You cannot migrate DataJoiner databases that contain abstract data types, such as DB2 Spatial Extender data, or extended indexes. The abstract data types or extended indexes must be removed before the database can be migrated.

Access to LOB columns in DB2 Universal Database for Linux, UNIX, and Windows, version 7 and DB2 Universal Database for z/OS, version 7

The following restrictions apply to applications that access LOB data types:

- In DataJoiner, the DB2RA protocol allowed read and write operations on LOB columns in DB2 Universal Database for Linux, UNIX, and Windows and DB2 Universal Database for z/OS, version 7 data sources. WebSphere Federation Server uses the DRDA protocol and does not support the DB2RA protocol. The DRDA protocol does not support LOB data types when accessing DB2 Universal Database for Linux, UNIX, and Windows and DB2 Universal Database for z/OS, version 7 data sources.
- DataJoiner applications that use the DB2RA protocol to perform operations on LOB columns in DB2 Universal Database for Linux, UNIX, Windows and DB2 UDB for z/OS data sources return errors when those applications are migrated to WebSphere Federation Server, version 9.1.

Access to LONG data types in Oracle tables

When you migrate Oracle nicknames to WebSphere Federation Server from DataJoiner, LONG data type columns will be read only.

In WebSphere Federation Server, the Oracle NET8 wrapper supports read and write operations on LOB columns. However, after you migrate to WebSphere Federated Server, LONG data type columns are read only. If you want to perform insert, update, and delete operations on LONG columns, use the Oracle utilities to alter the remote column data type to a LOB then recreate the nickname.

The following table shows the mappings from DB2 data types to Oracle LONG data types.

Table 2. DataJoiner mappings from DB2 data types to Oracle data types

DB2 data types	Oracle data types
<ul style="list-style-type: none"> • VARCHAR([256-4000]) FOR BIT DATA • LONG VARCHAR FOR BIT DATA • BLOB 	LONG RAW
<ul style="list-style-type: none"> • VARCHAR([2001-4000]) • LONG VARCHAR • CLOB 	LONG

When you migrate to WebSphere Federation Server, the local column data types in the catalog are changed. The local column data type for LONG VARCHAR columns is converted to the CLOB data type. The local column data type for LONG VARCHAR FOR BIT DATA columns is converted to the BLOB data type. The remote columns do not change.

AIX operating system support

If you use DataJoiner on AIX version 4.1, you must upgrade to AIX version 4.3.3 with maintenance level 9 or later before you migrate.

Install PTF 14 on DataJoiner Version 2.1.1 before you migrate to WebSphere Federation Server. For more information, see the download site for PTF 14 for DataJoiner for AIX.

Code page support

The code page support for DB2 V9.1 database systems might be different than the code page support for DataJoiner. For the code page support for D2 V9.1 database systems, see Supported territory codes and code pages.

Data access modules for Microsoft SQL Server

The DataJoiner migration to WebSphere Federation Server changes the default names of the data access modules for Microsoft SQL Server. Although the names are different, the wrapper links to the correct libraries and functions properly. The migration changes the following data access module names to DJXMSSQL3:

DataJoiner for Windows

mssqlodbc

DataJoiner for UNIX

djxmssql

djxsqlnk (DataDirect SequeLink ODBC driver)

WebSphere Federation Server supports two default wrapper names: DJXMSSQL3 and MSSQLODBC3. The default wrapper names are no longer associated with a specific operating system. You can use either default wrapper name when you create a Microsoft SQL Server wrapper. WebSphere Federation Server determines which operating system you are using and creates the correct wrapper library files for that operating system.

If you create a wrapper for Microsoft SQL Server, use the default wrapper name MSSQLODBC3. The default wrapper name DJXMSSQL3 is provided only to support migration from DataJoiner.

Database path location

DataJoiner, version 2.1.1 databases use a database path convention that is not supported in WebSphere Federation Server. The DataJoiner version 2.1.1 path is as follows:

```
user_specified_path/instance_name/SQLnnnnn
```

SQLnnnnn is the database subdirectory.

The path convention in WebSphere Federation Server is as follows:

```
user_specified_path/instance_name/NODExxxx/SQLnnnnn
```

NODExxxx is the node number, and SQLnnnnn is the database subdirectory.

All DataJoiner, version 2.1.1 databases are relocated from their original path to this new path. The database is automatically relocated during the database migration process.

DataDirect SequeLink ODBC driver

The DataDirect SequeLink ODBC driver for Microsoft SQL Server data sources is not supported in WebSphere Federation Server. For UNIX, you must use the DataDirect Technologies Connect ODBC version 4.2 or later driver.

You might need to update DataJoiner versions, modify your applications, alter data types, drop server options, and drop data type mappings.

DataJoiner Classic Connect data sources

WebSphere Federated Server does not support access to DataJoiner Classic Connect data sources such as IMS™ and VSAM. However, you can use WebSphere Information Integrator Classic Federation for z/OS™ to access these data sources. For information about how to access DataJoiner Classic Connect data sources, see the documentation for WebSphere Information Integrator Classic Federation for z/OS.

DBLIB wrapper

In WebSphere Federation Server, the DBLIB wrapper is not supported. If you use the DBLIB wrapper in DataJoiner, after you migrate to WebSphere Federation Server, you must use the CTLIB wrapper to access Sybase data sources.

You must submit the correct SQL statements to configure the CTLIB wrapper. You can use the db2look command to help you generate the proper SQL statements.

DB2 Run-time Client support

DataJoiner does not support connections from DB2 Universal Database, version 8 Run-time Clients or DB2 V9.1 Run-time Clients. If you connect to DB2 Universal Database, version 7 Run-time Clients or servers from DataJoiner, you must migrate from DataJoiner to DB2 Information Integrator, version 8.2, before you migrate the DB2 Universal Database, version 7 Run-time Clients and servers to DB2 Universal

Database version 8.2. For information about migrating Run-time Clients from DB2 Universal Database, version 8.2 to DB2 V9.1 database systems, see the DB2 Run-Time Client documentation.

LONG data types in DataJoiner nicknames

DB2 LONG data types are not currently supported in WebSphere Federation Server. When DataJoiner nickname columns of type LONG are migrated to WebSphere Federation Server, the local column data types in the federated database system catalog are changed to LOB data types. The following table shows the mapping for nickname column LONG data types after migration to WebSphere Federation Server.

Table 3. Data type mappings for LONG data types in WebSphere Federation Server

DataJoiner nickname data type	Federated data type
LONG VARCHAR	CLOB
LONG VARGRAPHIC	DBCLOB
LONG VARCHAR FOR BIT DATA	BLOB

Applications that perform insert and update operations on LONG data types might return errors when these applications are migrated to WebSphere Federation Server. Although the local column data types are converted to LOB data types, in WebSphere Federation Server, only the Oracle NET8 wrapper and the DB2 wrapper allow write operations on these columns when the remote type is LOB and not LONG. Remove write operations on LOB columns from applications that use a wrapper other than the Oracle NET8 wrapper or the DB2 wrapper.

If you need to perform insert and update operations on these local LOB data types, you might be able to alter these LOB data types to VARCHAR data types depending on the size of the data and on the data source that you are accessing.

Nicknames

In DataJoiner, version 2.1.1, you can create nicknames on DB2 aliases. However, you cannot create nicknames on DB2 aliases in WebSphere Federation Server.

Nonvalid data type mappings

DataJoiner does not check to determine if a data type mapping is valid when you create it. Instead, transparent DDL statements that use nonvalid data type mappings and queries that use nonvalid data type mappings fail. In WebSphere Federated Server, if you specify a server name with a type mapping or if you specify a server type that meets some existing server names, the validity of the data type mapping is checked when you attempt to create the mapping. You know immediately that the mapping is not valid because WebSphere Federation Server does not allow you to create the mapping.

Drop nonvalid data type mappings in DataJoiner before you migrate to WebSphere Federation Server.

Nonvalid server options

In DataJoiner, version 2.1.1, the DATEFORMAT, TIMEFORMAT, and TIMESTAMPFORMAT server options are available for remote servers for all

wrappers. However, these options are supported only for the ODBC wrapper in WebSphere Federation Server. You must drop these options before you migrate, if they are specified on a remote server that is created for wrappers other than the ODBC wrapper. For the ODBC wrapper, you can drop these server options on the remote server after you migrate to WebSphere Federation Server.

Read and write operations with large object (LOB) data types

The following restrictions apply to applications that access LOB data types:

- All relational wrappers allow read operations on LOB data type columns at the supported data sources.
- The Oracle NET8 wrapper allows read and write operations on LOB data type columns at Oracle data sources.
- The DB2 wrapper allows read and write operations on LOB data type columns on DB2 data sources from DB2 Universal Database for Linux, UNIX, and Windows, DB2 Universal Database iSeries, and DB2 Universal Database for z/OS and OS/390®.
- With the exception of the Oracle NET8 wrapper and the DB2 wrapper, WebSphere Federation Server currently does not support write operations on LOB data type columns. Applications that use other wrappers and perform insert, update, or delete operations on LOB data type columns return errors if the applications are migrated to WebSphere Federation Server.
- If you need to perform insert, update, or delete operations on local LOB data type columns, you might be able to alter the data types of these columns to VARCHAR data types depending on the size of the data and on the data source that you access.

Replication restrictions on LONG and LOB data types

When LONG data type information is migrated to WebSphere Federation Server, the local column data types in the system catalog for the federated database are changed by default to DB2 LOB data types: CLOB, DBCLOB, and BLOB. After the column data types of the nickname are changed from LONG data types to DB2 LOB data types, DB2 replication cannot replicate the LOB data.

If you are currently replicating local LONG data types that the migration process converts to LOB data types, you can alter the local LOB data type columns to VARCHAR data type columns after you migrate to WebSphere Federation Server. You can alter the column types to VARCHAR data types to allow insert, update, and delete privileges on these columns by DB2 replication, and these column values can be replicated.

You can check the local column data types in the DataJoiner system catalog to determine if LONG or LOB data type columns are present in the nickname for the target database object.

Spatial data

If your DataJoiner database contains spatial data, you must migrate this data separately from any non-spatial data.

Before migrating to WebSphere Federation Server, you must perform the following actions:

1. Manually export the spatial data tables to an SDE transfer file.

2. Drop the tables that contain spatial data columns.
3. Run the `disable_db` stored procedure to remove all of the catalog tables that refer to spatial data.

After the migration of the non-spatial data is complete, you must perform the following actions:

1. Enable the DB2 Spatial Extender database that you want to work with.
2. Import the spatial data into the federated database.

The commands to export and import data in previous versions of DB2 Spatial Extender changed. You must use the current syntax to import and export data. For the current syntax, see the DB2 Spatial Extender documentation.

SQLNET wrapper

In WebSphere Federation Server, the SQLNET wrapper is not supported. If you use the SQLNET wrapper in DataJoiner, after you migrate to WebSphere Federation Server, you must use the NET8 wrapper to access Oracle data sources.

You must submit the correct SQL statements to configure the NET8 wrapper. You can use the `db2look` command to help you generate the proper SQL statements.

SQL DDL statement and syntax changes

Some SQL DDL statements and syntax that are supported in DataJoiner are different in DB2 V9.1 database systems:

- The `CREATE SERVER MAPPING` statement is not supported in DB2 V9.1 database system. You can use the `CREATE WRAPPER` statement to register a wrapper with a federated server. You can use the `CREATE SERVER` statement to define a data source to a federated database.
- The `CREATE SERVER OPTION` is not supported in DB2 V9.1 database system. You can use the `ALTER SERVER` statement to modify the definition of a specific data source, modify the definition of a category of data sources, change the configuration of a specific data source, or change the configuration of a category of data sources.
- The `CREATE TABLE` statement (Transparent DDL) in DB2 V9.1 database system has a different syntax than the `CREATE TABLE` statement in DataJoiner.
- For the `CREATE TYPE MAPPING` statement in DB2 V9.1 database system, a `CHAR` data type that does not have a specified length is equivalent to `CHAR(1)`.
- The `CREATE USER MAPPING` statement is supported in DB2 V9.1 database system but uses a different syntax than the `CREATE USER MAPPING` statement in DataJoiner.
- The `CREATE VIEW` statement with the `WITH CHECK OPTION` clause is not supported in DB2 V9.1 database system.
- The `CREATE STORED PROCEDURE NICKNAME` statement is not supported in DB2 V9.1 database system. DB2 V9.1 database system uses the `CREATE PROCEDURE` statement to create federated stored procedures.

Stored procedure nicknames

Before you migrate applications from DataJoiner that invoke stored procedure nicknames, you must modify those applications. For DataJoiner applications that invoke a stored procedure nickname, you must drop the stored procedure

nickname and update the application to invoke a federated stored procedure that you create in WebSphere Federation Server. A federated stored procedure is a local stored procedure that is mapped to a stored procedure at a data source.

Federated stored procedures automatically determine the local data types by using forward type mappings that are applied to the remote data source object. In DataJoiner stored procedure nicknames, you can specify the local data types in the CREATE STORED PROCEDURE NICKNAME statement. Federated stored procedures do not allow you to specify different local data types for remote data types. For example, consider a remote procedure that has the arguments CHAR(5) and CHAR(5). In DataJoiner, you use the CREATE STORED PROCEDURE NICKNAME statement to change the local data types that correspond to remote procedure arguments. Federated stored procedures use the same forward data type mapping for the two CHAR(5) arguments, so the local data types are the same.

Triggers that reference nicknames

When you migrate from DataJoiner, triggers that reference nicknames are migrated but are not functional in WebSphere Federation Server. In some cases, you can drop and recreate the trigger in WebSphere Federation Server:

- If the triggered body contains only SELECT statements on nicknames, you can drop and recreate the trigger in WebSphere Federation Server, after you migrate.
- If the triggered body contains INSERT, UPDATE, or DELETE statements on nicknames, you cannot recreate the trigger in WebSphere Federation Server. You can drop these triggers in WebSphere Federation Server, after you migrate

Two-phase commit server option

The DataJoiner TWO_PHASE_COMMIT server option is not migrated when you migrate to WebSphere Federation Server. Before you enable a federated server to use the two-phase commit protocol, you must ensure that your target data source supports it and that the federated system can support the target data source. In WebSphere Federation Server, you can use the ALTER SERVER statement to add the DB2_TWO_PHASE_COMMIT server option and set a value for the server option.

If you migrate applications from DataJoiner that enable two-phase commit operations, you must update the existing applications by replacing the DataJoiner TWO_PHASE_COMMIT server option with the WebSphere Federation Server DB2_TWO_PHASE_COMMIT server option to enable federated two-phase commit operations.

User-defined type mapping on Oracle LONG columns

DataJoiner supports user-defined type mapping on nickname columns from VARCHAR to Oracle LONG data types. When you migrate a DataJoiner nickname that has a user-defined type mapping from VARCHAR to Oracle LONG, WebSphere Federation Server drops the user-defined type mapping but retains the local VARCHAR column in the nickname to allow write operations on the Oracle LONG column. However, if you create a new nickname on the remote Oracle table, WebSphere Federation Server maps a local LOB column to the Oracle LONG, so the nickname column is read only. To allow insert, update, a delete operations on the nickname column, you can use the ALTER NICKNAME statement to specify a local VARCHAR data type for the Oracle LONG.

Related concepts

About importing and exporting spatial data

Federated stored procedures

“Preparing to migrate to federated systems” on page 15

Before you migrate to IBM WebSphere Federation Server, you must perform tasks such as backing up your databases, recording system configurations, recording database and instance information, changing the diagnostic error level, and verifying that your databases are ready for migration.

Related tasks

“Allowing write operations on large object data types after migrating to WebSphere Federation Server” on page 34

After you migrate to IBM WebSphere Federation Server, the large object data types of your remote data sources are mapped to CLOB, BLOB, or DBCLOB data types, depending on the data source that you are accessing.

“Copying configuration settings” on page 15

Copy the environment variable configuration settings that are currently on your system. You must make a copy of the db2dj.ini file, so that you can restore the configuration settings after you migrate. If you plan to access Sybase data sources after you migrate, you might need to copy the Sybase interfaces file before you migrate.

Importing SDE transfer data to a new or existing table

Creating federated stored procedures

Related reference

db2look - DB2 statistics and DDL extraction tool command

WebSphere Information Integrator, version 8 migration restrictions

Before you migrate to IBM WebSphere Federation Server, you should be aware of migration restrictions when you migrate federated systems from WebSphere Information Integrator, version 8.

Local LOB columns that are mapped to Oracle LONG data types

WebSphere Information Integrator nicknames can contain local LOB columns that are mapped to Oracle LOB columns that are altered from Oracle LONG and LONG RAW data types. When you migrate the nickname that contains local LOB columns to WebSphere Federation Server, write operations on the local LOB columns that are mapped to Oracle LOB columns fail. If you want to perform INSERT, UPDATE, and DELETE operations on the Oracle LOB columns, you must drop the existing nickname and recreate the nickname with local LOB columns that map to the Oracle LOB columns.

Related concepts

“Preparing to migrate to federated systems” on page 15

Before you migrate to IBM WebSphere Federation Server, you must perform tasks such as backing up your databases, recording system configurations, recording database and instance information, changing the diagnostic error level, and verifying that your databases are ready for migration.

Related tasks

“Allowing write operations on large object data types after migrating to WebSphere Federation Server” on page 34

After you migrate to IBM WebSphere Federation Server, the large object data types of your remote data sources are mapped to CLOB, BLOB, or DBCLOB data types, depending on the data source that you are accessing.

“Copying configuration settings”

Copy the environment variable configuration settings that are currently on your system. You must make a copy of the db2dj.ini file, so that you can restore the configuration settings after you migrate. If you plan to access Sybase data sources after you migrate, you might need to copy the Sybase interfaces file before you migrate.

Preparing to migrate to federated systems

Before you migrate to IBM WebSphere Federation Server, you must perform tasks such as backing up your databases, recording system configurations, recording database and instance information, changing the diagnostic error level, and verifying that your databases are ready for migration.

To prepare for migration to WebSphere Federation Server, you might need to complete the following tasks:

- Copy configuration settings
- Back up databases before migration
- Record database information before migration
- Record instance information before migration
- Capture error and diagnostic information during migration
- Verify that your databases are ready for migration
- Take a DB2 database offline for migration

Related concepts

“Overview for migrating to WebSphere Federation Server” on page 1

You can migrate existing instances and databases to IBM WebSphere Federation Server, version 9.1 from supported versions of DB2 DataJoiner, DB2 database systems, and WebSphere Information Integrator, version 8. However, direct migration to WebSphere Federation Server, version 9.1 is supported only from WebSphere Information Integrator, version 8.2.

Related reference

Pre-migration tasks for DB2 servers

“DataJoiner migration restrictions” on page 7

Before you migrate from DataJoiner to IBM WebSphere Federation Server, there are migration restrictions and changes. You might have to perform tasks before or after you migrate, or you might have to limit certain database operations after migration.

“WebSphere Information Integrator, version 8 migration restrictions” on page 14

Before you migrate to IBM WebSphere Federation Server, you should be aware of migration restrictions when you migrate federated systems from WebSphere Information Integrator, version 8.

Copying configuration settings

Copy the environment variable configuration settings that are currently on your system. You must make a copy of the db2dj.ini file, so that you can restore the configuration settings after you migrate. If you plan to access Sybase data sources after you migrate, you might need to copy the Sybase interfaces file before you migrate.

Procedure

To copy the environment variable configuration settings from WebSphere Information Integrator, version 8.2:

- **Optional:** If you migrated databases only, make a copy of the db2dj.ini file so that you can restore the configuration settings after you migrate to IBM WebSphere Federation Server. If you migrate instances, the db2dj.ini file is migrated to WebSphere Federation Server. The db2dj.ini file contains configuration information about the data sources that you access. You can use the db2dj.ini file that you copy from WebSphere Information Integrator, version 8.2 to verify that the migration of your system configuration settings is successful after migrating to WebSphere Federation Server.
- **Optional:** If you plan to access Sybase data sources after you migrate to WebSphere Federation Server, and you can't use the default interfaces file (%SYBASE%\ini\sql.ini for Windows and \$SYBASE/interfaces for UNIX), copy the Sybase interfaces file. The interfaces file is typically located in the \$HOME/sqllib directory.

Related tasks

“Restoring configuration settings for federated systems” on page 29
You can restore the configuration settings after you migrate your instances and databases from WebSphere Information Integrator, version 8.2 to WebSphere Federation Server.

Related reference

“DataJoiner migration restrictions” on page 7
Before you migrate from DataJoiner to IBM WebSphere Federation Server, there are migration restrictions and changes. You might have to perform tasks before or after you migrate, or you might have to limit certain database operations after migration.

“WebSphere Information Integrator, version 8 migration restrictions” on page 14
Before you migrate to IBM WebSphere Federation Server, you should be aware of migration restrictions when you migrate federated systems from WebSphere Information Integrator, version 8.

Recording database information before migration

Record the database information and the node directory information before you migrate to IBM WebSphere Federation Server.

About this task

To record database information, use the LIST NODE DIRECTORY, LIST DATABASE DIRECTORY, and LIST DCS DIRECTORY commands.

Follow the steps in this task to preserve the necessary database information that you need, if a problem occurs during the instance migration.

Procedure

To record your database information:

1. Issue the LIST NODE DIRECTORY command. Record the node names, the host names, and the service names.
2. Issue the LIST DATABASE DIRECTORY command to list the contents of the system database directory. Record the database alias names. These names are the aliases for the databases that you access on remote servers.
3. Issue the GET DATABASE CONFIGURATION command to list information about database configuration parameters.

4. Issue the following statement when you migrate from DB2 Universal Database, version 8:

```
SELECT SERVERNAME,  
       SERVERTYPE,  
       OPTION,  
       SETTING  
FROM SYSCAT.SERVEROPTIONS  
WHERE OPTION IN ('NODE','DBNAME')  
ORDER BY SERVERNAME, OPTION;
```

Record the server names, node names, and database alias names of each DB2 family server definition that you cataloged.

Related tasks

“Recording instance information before migration”

Record the instance information before you migrate to WebSphere Federation Server. If a problem occurs during the instance migration, you need this information.

Verifying that your databases are ready for migration

Related reference

GET DATABASE CONFIGURATION Command

Recording instance information before migration

Record the instance information before you migrate to WebSphere Federation Server. If a problem occurs during the instance migration, you need this information.

Before you begin

On UNIX, you need root authorization to perform this task.

Procedure

To record instance information:

- Issue the db2ilist command at the operating system prompt. This command lists all the instances that are available on your system.
- Issue the GET DATABASE MANAGER CONFIGURATION command. This command lists information for the database manager configuration parameters.

Related tasks

“Recording database information before migration” on page 16

Record the database information and the node directory information before you migrate to IBM WebSphere Federation Server.

Verifying that your databases are ready for migration

Chapter 2. Migrating Windows servers

For Windows, you can migrate databases and instances to WebSphere Federation Server on 32-bit and 64-bit systems.

Migrating to WebSphere Federation Server on Windows

You can migrate directly from IBM WebSphere Information Integrator, version 8.2 on a 32-bit or 64-bit system to WebSphere Federation Server. For Windows, you can migrate databases and instances to WebSphere Federation Server on 32-bit and 64-bit systems.

Before you begin

Depending on the product you are migrating from, you might need to complete these tasks before you can migrate your instances and databases:

- If you are migrating from DataJoiner, version 2.1.1 or DB2 Universal Database, version 7, you must complete the following tasks:
 1. Follow the steps described in the DB2 Information Center, version 8.2 to migrate instances and databases from DataJoiner, version 2.1.1 or DB2 Universal Database, version 7 to WebSphere Information Integrator, version 8.2.
 2. Rebind packages.
 3. Optional: Run the db2uiddl command to convert DataJoiner unique indexes to the DB2 version 5 semantics that WebSphere Federation Server supports. The db2uiddl command generates CREATE UNIQUE INDEX statements for unique indexes on user tables.
 4. Set up access to the data sources.
- If you are migrating directly from WebSphere Information Integrator, version 8.2, you must complete tasks before you migrate:
 - Install the latest fix pack of WebSphere Information Integrator, version 8.2 on your system.
 - Prepare to migrate to WebSphere Federation Server.

Procedure

To migrate to WebSphere Federation Server:

1. Log on as a Local Administrator.
2. Install WebSphere Federation Server on a 32-bit or 64-bit computer. Do not uninstall WebSphere Information Integrator, version 8.2. Use the steps for setting up the server for the data sources that you want to access.

3. Migrate instances and databases or only databases:

Option	Description
<p>To migrate instances and databases:</p>	<p>For successful instance migration, all files for all instances must exist and write access must be granted.</p> <ol style="list-style-type: none"> 1. Run the <code>db2ckmig</code> command from the WebSphere Federation Server CD to verify that the database can be migrated. For example, <pre>cd\db2\Windows\utilities\db2ckmig database -l filename</pre> <p><i>cd</i> represents the mount point of the WebSphere Federation Server CD. <i>database</i> is the alias name of a database that you want to scan and <i>-l filename</i> specifies a log file for errors and warnings that are generated for the scanned database.</p> 2. Run the <code>db2imigr</code> command to migrate the instance to a DB2 version 9.1 database system. For example, <pre>"%DB2PATH%" \instance\db2imigr instance_name /u username, password</pre> <p>DB2PATH is set to the path that you specify during installation. <i>instance_name</i> is the name of the instance that you want to migrate, and <i>/u username, password</i> specifies the account name and password for the DB2 service. This option is required when you migrate a partitioned instance.</p> 3. Run the <code>MIGRATE DATABASE</code> command to migrate the DB2 Universal Database, version 8 database. For example, <pre>MIGRATE DATABASE database_name</pre> <p>where <i>database_name</i> is the name of the database you want to migrate.</p> <p>When the database migration is called implicitly using the <code>RESTORE DATABASE</code> command from a DB2 Universal Database, version 8 backup or explicitly using the <code>MIGRATE DATABASE</code> command, the following database entities are converted during database migration:</p> <ul style="list-style-type: none"> • Database configuration file • Log file header • Catalog tables • Buffer pool files • Index root page • History file

Option	Description
<p>To migrate databases only:</p>	<p>You can migrate individual databases after you migrate to WebSphere Federation Server, version 9.1.</p> <ol style="list-style-type: none"> 1. Run the db2ckmig command from the WebSphere Federation Server CD to verify that the database can be migrated. For example, <pre>cd\db2\Windows\utilities\db2ckmig database -l filename</pre> <i>cd</i> represents the mount point of the WebSphere Federation Server CD. <i>database</i> is the alias name of a database that you want to scan and <i>-l filename</i> specifies a log file for errors and warnings that are generated for the scanned database. 2. Backup databases on DB2 Universal Database, version 8.2 by using the BACKUP DATABASE command. 3. Run the RESTORE DATABASE command to restore the database on the DB2 V9.1 database system. <p>When the database migration is called implicitly using the RESTORE DATABASE command from a DB2 Universal Database, version 8 backup or explicitly using the MIGRATE DATABASE command, the following database entities are converted during database migration:</p> <ul style="list-style-type: none"> • Database configuration file • Log file header • Catalog tables • Buffer pool files • Index root page • History file

4. Verify the migration of your databases
5. If you migrated from DataJoiner, you must drop any existing stored procedure nicknames and recreate federated stored procedures in WebSphere Federation Server by using the CREATE PROCEDURE statement.

Related concepts

“Overview for migrating to WebSphere Federation Server” on page 1
You can migrate existing instances and databases to IBM WebSphere Federation Server, version 9.1 from supported versions of DB2 DataJoiner, DB2 database systems, and WebSphere Information Integrator, version 8. However, direct migration to WebSphere Federation Server, version 9.1 is supported only from WebSphere Information Integrator, version 8.2.

Related tasks

- Migrating a DB2 server (Windows)
- Creating federated stored procedures
- db2uiddl - Prepare unique index conversion to V5 semantics command

Chapter 3. Migrating Linux and UNIX servers

For AIX and Solaris, you can migrate databases and instances to WebSphere Federation Server on 64-bit systems only. For Linux, you can migrate databases and instances to WebSphere Federation Server on 32-bit and 64-bit systems.

Migrating to WebSphere Federation Server on Linux and UNIX

You can migrate directly from IBM WebSphere Information Integrator, version 8.2 on a 32-bit or 64-bit system to WebSphere Federation Server. For AIX and Solaris, you can migrate databases and instances to WebSphere Federation Server on 64-bit systems only. For Linux, you can migrate databases and instances to WebSphere Federation Server on 32-bit and 64-bit systems.

Before you begin

Depending on the product you are migrating from, you might need to complete these tasks before you can migrate your instances and databases:

- If you are migrating from DataJoiner, version 2.1.1 or DB2 Universal Database, version 7, you must complete the following tasks:
 1. Follow the steps described in the DB2 Information Center, version 8.2 to migrate instances and databases from DataJoiner, version 2.1.1 or DB2 Universal Database, version 7 to WebSphere Information Integrator, version 8.2.
 2. Rebind packages.
 3. Optional: Run the `db2uiddl` command to convert DataJoiner unique indexes to the DB2 version 5 semantics that WebSphere Federation Server supports. The `db2uiddl` command generates `CREATE UNIQUE INDEX` statements for unique indexes on user tables.
 4. Set up access to the data sources.
- If you are migrating directly from WebSphere Information Integrator, version 8.2, you must complete the tasks before you migrate:
 - Install the latest fix pack of WebSphere Information Integrator, version 8.2 on your system.
 - Prepare to migrate to WebSphere Federation Server.

Procedure

To migrate to WebSphere Federation Server:

1. Install WebSphere Federation Server on a 32-bit or 64-bit computer. Use the steps for setting up the server for the data sources that you want to access. Do not uninstall WebSphere Information Integrator, version 8.2.

2. Migrate instances and databases or only databases.

Option	Description
<p>To migrate instances and databases:</p>	<p>For instance migration, all files for all instances must exist and write access must be granted.</p> <ol style="list-style-type: none"> 1. Run the <code>db2ckmig</code> command as the instance owner to verify that the database can be migrated. For example, <pre>\$DB2DIR/instance/db2ckmig database -l filename</pre> <p><code>DB2DIR</code> is set to the path you specify during installation. The default installation path is <code>/opt/ibm/wsii/V9</code> on Linux and <code>/opt/IBM/wsii/V9</code> on UNIX. <code>database</code> is the alias name of a database that you want to scan and <code>-l filename</code> specifies a log file for errors and warnings that are generated for the scanned database.</p> 2. Run the <code>db2imigr</code> command to migrate the instance to a DB2 version 9.1 database system. For example, <pre>\$DB2DIR/instance/db2imigr -u FencedID instance_name</pre> <p><code>FencedID</code> is the name of the user ID under which fenced user-defined functions and fenced stored procedures will run.</p> 3. Run the <code>MIGRATE DB database_name</code> to migrate the DB2 Universal Database, version 8 database, where <code>database_name</code> is the name of the database you want to migrate. <p>When the database migration is called implicitly using the <code>RESTORE DATABASE</code> command from a DB2 Universal Database, version 8 backup or explicitly using the <code>MIGRATE DATABASE</code> command, the following database entities are converted during database migration:</p> <ul style="list-style-type: none"> • Database configuration file • Log file header • Catalog tables • Buffer pool files • Index root page • History file

Option	Description
<p>To migrate databases only:</p>	<p>You can migrate individual databases after you migrate your instances to WebSphere Federation Server, version 9.1.</p> <ol style="list-style-type: none"> 1. Run the \$DB2DIR/instance/db2ckmig command as the instance owner to verify that the database can be migrated. 2. Backup databases on DB2 Universal Database, version 8.2 by using the BACKUP DATABASE command. 3. Run the RESTORE DATABASE command to restore the database on the DB2 V9.1 database system. <p>When the database migration is called implicitly using the RESTORE DATABASE command from a DB2 Universal Database, version 8 backup or explicitly using the MIGRATE DATABASE command, the following database entities are converted during database migration:</p> <ul style="list-style-type: none"> • Database configuration file • Log file header • Catalog tables • Buffer pool files • Index root page • History file

3. If you migrated from DataJoiner, you must drop any existing stored procedure nicknames and recreate federated stored procedures in WebSphere Federation Server by using the CREATE PROCEDURE statement.

Related concepts

“Overview for migrating to WebSphere Federation Server” on page 1
 You can migrate existing instances and databases to IBM WebSphere Federation Server, version 9.1 from supported versions of DB2 DataJoiner, DB2 database systems, and WebSphere Information Integrator, version 8. However, direct migration to WebSphere Federation Server, version 9.1 is supported only from WebSphere Information Integrator, version 8.2.

Related tasks

Migrating a DB2 server (Linux and UNIX)

“Migrating 32-bit federated systems to 64-bit federated systems on UNIX” on page 26

You can migrate federated databases from WebSphere Information Integrator, version 8.2 on 32-bit systems to WebSphere Federation Server, version 9.1 on 64-bit systems. Migration from 32-bit federated systems to 64-bit federated systems is supported on AIX and Solaris.

Creating federated stored procedures

db2uiddl - Prepare unique index conversion to V5 semantics command

Migrating 32-bit federated systems to 64-bit federated systems on UNIX

You can migrate federated databases from WebSphere Information Integrator, version 8.2 on 32-bit systems to WebSphere Federation Server, version 9.1 on 64-bit systems. Migration from 32-bit federated systems to 64-bit federated systems is supported on AIX and Solaris.

Before you begin

- Verify that your databases are ready for migration.
- Backup your databases.
- You must have root access.
- You must have SYSADM, SYSCTRL, or SYSMAINT authority on the databases.
- The kernel of the operating system must be running in 64-bit mode.
- All data source client software that you require is installed and configured on the 64-bit system.

Restrictions

- You cannot revert to a DB2 Universal Database, version 8.2 32-bit instance after you migrate to a DB2 V9.1 64-bit instance.

Procedure

To migrate from a WebSphere Information Integrator 32-bit system to a WebSphere Federation Server 64-bit system:

1. Log in as the root user.
2. Run the `db2stop` command to stop the DB2 Universal Database, version 8.2 instance
3. **Optional:** Update the 32-bit instance to a 64-bit instance by issuing the `db2iupdt` command with the `-w` parameter set to 64. For example,
`DB2DIR/instance/db2iupdt -w 64 instance_name`

where `DB2DIR` is the DB2 Universal Database, version 8.2 installation path and `instance_name` is the name of the instance that you want to migrate.

Step 3 is recommended only if you are also migrating to 64-bit applications.

After migration to WebSphere Federation Server, all your instances are migrated to 64-bit instances, and `$INSTHOME/sqllib/lib` points to 64-bit shared libraries. You should test your applications to ensure the run successfully. If you do not update to a 64-bit instance, `$INSTHOME/sqllib/lib` points to 32-bit shared libraries.

4. Install WebSphere Federation Server on the 64-bit system. Do not remove WebSphere Information Integrator, version 8.2.
5. Run the `db2imigr` command to migrate the instance to a DB2 version 9.1 database system. For example,
`DB2DIR/instance/db2imigr -u FencedID instance_name`

`FencedID` is the name of the user ID under which fenced user-defined functions and fenced stored procedures run. `instance_name` is the name of the instance that you want to migrate.

6. Migrate databases.
7. Restart the instance by using the `db2start` command.

8. Verify the migration of your databases.

Related tasks

Migrating a DB2 server (Linux and UNIX)

Chapter 4. Accessing data sources after migration

After you migrate to WebSphere Federation Server, you must configure the federated server before you can access data sources.

Configuring federated systems after migration

After you migrate to WebSphere Federation Server, you can restore configuration settings, rebind packages, convert indexes, and so on, before you configure the federated server for the data sources you want to access.

You might need to complete the following tasks to configure the federated server after migration:

- Restore federated systems configuration settings
- Convert type-1 indexes to type-2 indexes
- Rebind packages
- Change the wrapper for a data source
- Alter the data source version in a server definition (DB2 Control Center)
- Alter the data source version in a server definition (DB2 command line)
- Set isolation-level behavior for applications after migration
- Restore connections to DB2 family data sources
- Allow write operations on large object data types after migrating to WebSphere Federation Server
- Access XML data sources after migrating to WebSphere Federation Server

Restoring configuration settings for federated systems

You can restore the configuration settings after you migrate your instances and databases from WebSphere Information Integrator, version 8.2 to WebSphere Federation Server.

Procedure

To restore the configuration settings for a federated system, restore the copy of the `db2dj.ini` that you created before you migrated to IBM WebSphere Federation Server. The `db2dj.ini` file is located in the file that the DB2 registry variable `DB2_DJ_INI` specifies. When the `DB2_DJ_INI` variable is not set, the default path is as follows:

- UNIX: `instancehome/sqllib/cfg/db2dj.ini`, where *instancehome* is the home directory of the instance owner.
- Windows: `%DB2PATH%\cfg\db2dj.ini`

where `%DB2PATH%` is the directory where the DB2 database system is installed. For example, `C:\Program Files\IBM\sqllib`.

Related tasks

“Copying configuration settings” on page 15

Copy the environment variable configuration settings that are currently on your system. You must make a copy of the `db2dj.ini` file, so that you can restore the configuration settings after you migrate. If you plan to access Sybase data sources after you migrate, you might need to copy the Sybase interfaces file before you migrate.

Converting type-1 indexes to type-2 indexes

To minimize next-key locking, convert type-1 indexes to type-2 indexes for a local table in a IBM WebSphere Federation Server database.

About this task

After you migrate to WebSphere Federation Server, you must convert type-1 indexes to type-2 indexes for a local table in a WebSphere Information Integrator database.

Procedure

To convert type-1 indexes to type-2 indexes:

1. To determine the index type, issue the following command:

```
db2 load query table [table_name]
```

table_name is the name of the table for which you want to determine the index type.

If the index is type-1, the output for the table is:

```
Tablestate: Normal
```

```
Type-1 Indexes
```

If the index is type-2, the output for the table is:Tablestate: Normal

You do not need to convert a type-2 index.

2. To convert the index to type-2, issue the following command:

```
db2 reorg indexes all for table <table_name>  
allow [write | read | no] access convert
```

As an alternative, you can drop all indexes on the table and create them again.

Related concepts

Index structure

Related tasks

db2uiddl - Prepare unique index conversion to V5 semantics command

Rebinding packages

Rebinding is the process of recreating a package for an application program that was previously bound. You must rebind packages after you migrate. The REBIND command uses the same bind options that were specified when the package was created.

Procedure

Issue the REBIND command to rebind packages:

```
REBIND package_name
```

where *package_name* is the name of the package you want to rebind.

If you do not rebind the packages, you might receive an error message when you run packages.

Related reference

REBIND command

Changing the wrapper for a data source

You can change the wrapper that you use to access a data source after you migrate. You might change a wrapper when you want to upgrade for new wrapper functionality.

About this task

DB2 Information Integrator, version 8.1 supported more than one wrapper for some data sources, such as the SQLNET and NET8 wrappers for Oracle data sources and the DBLIB and CTLIB wrappers for Sybase data sources. For WebSphere Information Integrator, version 8.2 and IBM WebSphere Federation Server, the SQLNET wrapper is not supported for Oracle data sources and the DBLIB wrapper is not supported for Sybase data sources.

You can change the wrapper by either reissuing all the SQL statements manually, or you can use the db2look command to obtain the correct syntax necessary for each data source configuration statement.

Tip: The db2look command generates DDL statements from the metadata in the DB2 catalogs. If you use the db2look command, you can edit the statements with your specific information.

Procedure

To change the wrapper that you use:

1. Run the db2look command to create the necessary DDL statements from DB2 Universal Database. For example:

```
db2look -d database_name -wrapper wrapper_name -e -o federated.sql
```

This command creates an output file named federated.sql that contains the DDL statements for all the federated objects of the wrapper that you specify.

2. Edit the SQL statement in the federated.sql file to specify the new wrapper information.
3. If the wrapper that you plan to drop accesses remote tables that you created by using transparent DDL, create nicknames again for the remote tables. You can add the SQL statements for these nicknames to the federated.sql file.
4. Drop the wrapper.
5. Register the new wrapper with the federated database.
6. Add the client paths to the db2dj.ini file:
 - a. Edit the db2dj.ini file to include the client paths for the data source that you are using.
 - b. Issue the dxlink script to create new libraries in the DB2 base directories.
 - c. Issue the db2stop command.
 - d. Issue the db2iupdt command to update your instances.
 - e. Issue the db2start command to restart the DB2 instance.
7. Issue the SQL statements in the federated.sql file to configure access to the data source. For example:

```
db2 -tvf federated.sql
```

8. Bind any application packages that were invalidated when you dropped the wrapper. Packages that can be invalidated contain static SQL that references nicknames that were dropped and recreated.

9. Recreate any federated views that were invalidated when you dropped the wrapper. Views that were invalidated are the views that reference nicknames that were dropped and recreated.

Related tasks

Dropping a wrapper

Related reference

Restrictions for the db2dj.ini file

Specifying wrapper information in the federated.sql file

When you use the db2look command to change the wrapper that you use to access a data source, you need to edit the SQL statements in the federated.sql file.

Procedure

To edit the SQL statements in the federated.sql file:

1. Ensure that the SQL statements in the federated.sql file are in the following order:
 - a. Create the wrapper.
 - b. Create the server definitions.
 - c. Create the user mappings.
 - d. Create the nicknames, including nicknames for remote tables that were created using transparent DDL.
 - e. Alter the nicknames to override the default mappings.
2. Change the wrapper name. The wrapper name is on both the CREATE WRAPPER and CREATE SERVER statement.
 - Use the default wrapper name so that you do not have to specify the library file name.
 - If you do not use the default wrapper name, ensure that the wrapper library name is correct. The wrapper libraries are in the file system. On UNIX systems, the wrapper libraries are in the *instancehome*/sqllib/lib directory where *instancehome* is the home directory of the instance owner. On Windows systems, the wrapper libraries are in the %DB2PATH%\SQLLIB\BIN directory where %DB2PATH% is the directory where DB2 is installed, for example, C:\Program Files\IBM\sqllib. The CREATE WRAPPER statement must specify the base library of the wrapper, which is the library name. The base library name does not include, 'F' (fenced) or 'U' (unfenced).
3. If the AUTHORIZATION and PASSWORD parameters are used with the CREATE SERVER statement, the values that are associated with these parameters are empty in the SQL statements generated by the db2look command. You must add the correct values for these parameters.
4. Manually enclose the remote data types in the CREATE TYPE MAPPING statements in double quotation marks (") if you want the characters to be case sensitive. The remote data types in the generated CREATE TYPE MAPPING statements are not enclosed with double quotation marks.
5. Some data type mappings for certain data sources in WebSphere Federation Server are not the same as they were in DB2 Information Integrator, version 8.1. You can now use empty parentheses for parameterized data types. When you run db2look on a migrated database, empty parentheses are added not only against parameterized data types that do not have length but also against non-parameterized data types. If you have non-parameterized data types, remove the empty parentheses.
6. Insert the correct passwords in the CREATE USER MAPPING statements.

7. Check the SQL statements for any local views that are created by using the WITH CHECK OPTION clause. This clause is no longer supported. Remove the SQL statements that create these local views from the federated.sql file.

Related reference

db2look - DB2 statistics and DDL extraction tool command

CREATE WRAPPER statement

CREATE SERVER statement

Setting isolation-level behavior for applications after migration

WebSphere Federation Server supports statement-level isolation for remote DB2 family data sources and Microsoft SQL Server data sources. After you migrate to WebSphere Federation Server, existing applications that contain a WITH isolation clause in SELECT, INSERT, UPDATE and DELETE statements might show different isolation behavior. If you want the application to show premigration isolation level behavior, you must disable statement-level isolation.

Procedure

To disable statement isolation levels for a data source server:

Set the DB2_STATEMENT_ISOLATION server option to “N” from the CREATE SERVER or ALTER SERVER statement, for example:

```
ALTER SERVER db2server OPTIONS (ADD DB2_STATEMENT_ISOLATION 'N')
```

Related reference

Maintain data integrity with isolation levels

Statement level isolation in a federated system

Connection level isolation in a federated system

Configuring XML data sources after migration

When you migrate to IBM WebSphere Federation Server, you might need to alter the nickname column names or alter other objects.

About this task

The default column names generated for XML nicknames by the discover feature in the DB2 Control Center changed in DB2 Universal Database, version 8.2 and later. If you drop a nickname for an XML data source that was created in a release prior to version 8.2 and then create the nickname again in WebSphere Federation Server, the default column names will be different. You might need to alter the nickname column names or alter other objects, such as your SQL queries, that refer to the old column names.

Procedure

To alter nickname column names, see Altering nickname column names (DB2 Control Center).

Related tasks

Altering a nickname

Altering a local type for a data source object – examples

Restoring connections to DB2 family data sources

To access a remote DB2 family data source after you migrate to WebSphere Federation Server, you use the BIND command to restore a connection from the server to the remote data source.

Procedure

To restore the connection to the data source:

1. Connect to the remote DB2 family data source.
2. Issue the following commands:

- UNIX:

```
cd instancehome/sql1lib/bnd
DB2 BIND @db2cli.1st GRANT public
```

where *instancehome* is the home directory of the instance owner.

- Windows:

```
cd %DB2PATH%\bnd
DB2 BIND @db2cli.1st GRANT public
```

where %DB2PATH% is the directory where the DB2 database system is installed, for example, C:\Program Files\IBM\sql1lib.

3. Disconnect from the remote data source.

Repeat the entire procedure for each remote DB2 family data source that you want to access.

Allowing write operations on large object data types after migrating to WebSphere Federation Server

After you migrate to IBM WebSphere Federation Server, the large object data types of your remote data sources are mapped to CLOB, BLOB, or DBCLOB data types, depending on the data source that you are accessing.

About this task

With the exception of the Oracle NET8 wrapper and the DRDA wrapper, federated servers cannot write to DB2 CLOB, BLOB, or DBCLOB data types in nicknames.

If you need to write to these large object data types after you migrate to WebSphere Federation Server, you can alter the local CLOB and BLOB data type columns to VARCHAR and VARCHAR FOR BIT DATA data types.

You can alter CLOB and BLOB data types to VARCHAR data types with lengths up to 32672 bytes. If the CLOB, BLOB, or DBCLOB data type contains data that is larger than 32672 bytes, the data is truncated.

To enable replication of large object data types, you must allow write operations on these large object data type columns. DB2 replication requires insert and update privileges on the column values that it replicates to target tables.

Procedure

To allow write operations on large object data types:

Issue the ALTER NICKNAME statement. For example:

```
ALTER NICKNAME nickname1
ALTER COLUMN column1
LOCAL TYPE VARCHAR(32672)
```

You might receive SQL statement warnings when you attempt to alter large object data types depending on the size of the data and the data source that you are accessing.

Related reference

“DataJoiner migration restrictions” on page 7

Before you migrate from DataJoiner to IBM WebSphere Federation Server, there are migration restrictions and changes. You might have to perform tasks before or after you migrate, or you might have to limit certain database operations after migration.

“WebSphere Information Integrator, version 8 migration restrictions” on page 14

Before you migrate to IBM WebSphere Federation Server, you should be aware of migration restrictions when you migrate federated systems from WebSphere Information Integrator, version 8.

Default forward data type mappings

Default reverse data type mappings

Write operations on large object data types

For each data source that IBM WebSphere Federation Server supports, the wrapper maps specific remote data types to local large object data types by default.

The following table shows the default data type mappings for WebSphere Federation Server data sources from remote data types to local large object data types. You can allow write operations on and enable replication of large object data types for the data sources by altering the local CLOB, BLOB, and DBCLOB data type columns to VARCHAR and VARCHAR FOR BIT DATA data types with lengths up to 32672 bytes.

Table 4. Remote data types that are mapped to local large object data types for WebSphere Federation Server data sources

Data source	Remote data type	Local data type
DB2 family	LONG VARCHAR and LONG VARCHAR FOR BIT DATA	CLOB, BLOB, and DBCLOB
Oracle (NET8 wrapper)	LONG and LONG RAW	CLOB, BLOB, and DBCLOB
Informix	TEXT and BYTE	CLOB, BLOB, and DBCLOB
Sybase (CTLIB wrapper)	TEXT and IMAGE	CLOB and BLOB
Microsoft SQL Server (DjxMSSQL3 wrapper and MSSQLODBC3 wrapper)	TEXT and IMAGE	CLOB, BLOB, and DBCLOB
Teradata	CHAR and VARCHAR with lengths > 32672	CLOB
Teradata	BYTE and VARBYTE with lengths > 32672	BLOB
Teradata	GRAPHIC and VARGRAPHIC with lengths >16336	DBCLOB

Consider a remote table that is named EMP that references a remote Informix data source. The EMP table contains a column named EMP_INFO with a remote data type of LONG VARCHAR. To use the remote table, create a nickname that is named EMPLOYEE that references the remote Informix table.

When you create a nickname for this remote table, the Informix wrapper maps the data type of the EMP_INFO column by default to a local CLOB data type.

To allow write operations on and enable replication of the local data in the EMP_INFO column, you can alter the data type of the column from CLOB to VARCHAR. For example:

```
ALTER NICKNAME EMPLOYEE
ALTER COLUMN EMP_INFO
LOCAL TYPE VARCHAR(32672)
```

Accessing DB2 family data sources after migrating to WebSphere Federation Server

When you migrate to WebSphere Federation Server from WebSphere Information Integrator, version 8.2, you might need to configure the federated server to access DB2 family data sources. If you migrated to WebSphere Information Integrator, version 8.2 from DataJoiner or DB2 Universal Database version 7 federated servers, and then migrated only databases to WebSphere Federation Server, you must complete the following procedure to access DB2 family data sources.

Before you begin

Obtain the server names and the database alias names for the databases that you accessed before you migrated to WebSphere Federation Server.

Although the DRDA wrapper for WebSphere Federation Server ignores the NODE option, a node entry must be cataloged in the node directory to catalog remote databases in the database directory.

About this task

DataJoiner and DB2 Universal Database, version 7 federated servers use the NODE option to match a local node directory entry with the DBNAME option. These options match a database alias, location, or remote database name at the remote data source. WebSphere Information Integrator, version 8 and WebSphere Federation Server use the DBNAME option, which matches a database alias in the local database directory.

Procedure

To access DB2 family data sources after you migrate to WebSphere Federation Server:

1. Issue the LIST DATABASE DIRECTORY command to display the names of the databases that are defined in the database directory for the federated server system.
2. Issue the LIST NODE DIRECTORY command to display the contents of the node directory.
3. Verify that the remote node names from step 1 are the valid node entries in the node directory from step 2.

4. If you migrated only the database, issue the CATALOG TCPIP NODE command. This command adds the TCP/IP node entry to the node directory so that you can access the remote node. For example:

```
CATALOG TCPIP NODE NODENAME1 REMOTE hostname SERVER servicename
```

NODENAME1 is the name of the remote node. *hostname* is the host name where the target database resides. *servicename* is the service name or port number of the database manager instance for the server.

5. Catalog the database:
 - a. Issue the CATALOG DATABASE command to catalog and store remote database information in the directory for the federated system database. For example:

```
CATALOG DATABASE DB_NAME AS ALIAS1  
AT NODE NODENAME1
```

DB_NAME is the name of the remote database that you are cataloging in the database directory for the federated server system. *ALIAS1* is the alias for the database that you are cataloging. *NODENAME1* is the name of the node where this remote database resides. The name of the node that you specify must match the node name that you added in the CATALOG TCPIP NODE command.

- b. If the remote database name is longer than eight characters, issue the CATALOG DCS DATABASE command. This command creates a DCS directory entry. For example:

```
CATALOG DCS DATABASE ALIAS1 AS DB2_DATABASE_NAME
```

The name of the alias, *ALIAS1* in this example, must match the name of the database alias that you are cataloging. *DB2_DATABASE_NAME* is the name of the target host database. Database names for DB2 UDB for z/OS and DB2 UDB for iSeries data sources are typically longer than eight characters.

6. Alter the server definition in the system catalog for the federated database to register the database name:

- a. Check the SYSCAT.SERVERS catalog view for the name of the server that you migrated.

- b. Issue the ALTER SERVER statement. For example:

```
ALTER SERVER alias_server_name OPTIONS  
(SET DBNAME 'ALIAS1')
```

alias_server_name is the alias for the remote server in the catalog view.

ALIAS1 is the alias for the database that you cataloged.

7. Verify that the migration was successful:

- a. Connect to the database.
 - b. Issue SELECT, INSERT, UPDATE, and DELETE statements on the nicknames to verify that you can view and modify the data. You can also open a pass-through session to the remote database and issue a SELECT statement on a remote table.

Accessing Informix data sources after migrating to WebSphere Federation Server

To access Informix data sources if you migrated the instance and database or only the database, you set environment variables, run scripts, and verify the migration.

About this task

If you migrated only the database, follow all of the steps in this procedure.

If you migrated the instance and database on UNIX, you might need to perform only some of the steps in this procedure:

- Set the `INFORMIXDIR` environment variable.
- Run the `djxlink` script or the `djxlinkInformix` script.
- Recycle the DB2 instance.
- Verify the migration.

Procedure

To access Informix data sources after you migrate to WebSphere Federation Server:

1. Verify the connection to the Informix server. The connection information is in the `sqlhosts` file:
 - UNIX: The configuration information is in the file that the `INFORMIXSQLHOSTS` environment variable specifies. If `INFORMIXSQLHOSTS` is not specified, the default path is `$INFORMIXDIR/etc/sqlhosts`.
 - Windows: The `sqlhosts` information is in the Windows registry under the key `HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS`. You can set the `INFORMIXSQLHOSTS` environment variable to the name of the Windows computer that stores the registry. The default is the local computer.
2. Check the settings of the Informix environment variables:
 - a. Find the `db2dj.ini` file. The `db2dj.ini` file is located in the file that is specified by DB2 registry variable, `DB2_DJ_INI`. When this variable is not set, the default is as follows:
 - UNIX:
`instancehome/sql1lib/cfg/db2dj.ini`

where `instancehome` is the home directory of the instance owner.
 - Windows:
`%DB2PATH%\cfg\db2dj.ini`

where `%DB2PATH%` is the directory where the DB2 database system is installed, for example, `C:\Program Files\IBM\sql1lib`.
 - b. Set the environment variables in the `db2dj.ini` file as necessary:

INFORMIXDIR

The fully-qualified directory path where the Informix Client SDK software is installed. For example:

```
INFORMIXDIR=/informix/csdk
```

INFORMIXSERVER

The name of the default Informix server. For example:

```
INFORMIXSERVER=inf10
```

The INFORMIXSERVER setting must be a valid entry in the sqlhosts file on UNIX or the SQLHOSTS registry key on Windows. To get a value for the INFORMIXSERVER variable, look in the sqlhosts file. Select one of the values. The *dbservername* is the first value in each entry in the sqlhosts file. Although the Informix wrapper does not use the value of this variable, the Informix client requires that this variable be set.

INFORMIXSQLHOSTS

The path where the sqlhosts file resides. Set the INFORMIXSQLHOSTS environment variable only if the path that you are using for the sqlhosts file is different from the default path. The following example shows you how to set the INFORMIXSQLHOSTS variable to another path on UNIX:

```
INFORMIXSQLHOSTS=/informix/csdk/etc/my_sqlhosts
```

On Windows, if the SQLHOSTS registry key does not reside on the local computer, then the INFORMIXSQLHOSTS value is the name of the Windows computer that stores the registry.

- c. Set the following environment variables, if you want to change the default values for Informix code page conversion:
 - Set the CLIENT_LOCALE environment variable to the Informix locale. For example:

```
CLIENT_LOCALE=informix_client_locale_value
```
 - Set the DB_LOCALE environment variable to the name of the Informix database locale. For example:

```
DB_LOCALE=informix_db_locale_value
```

Set this environment variable if the Informix database uses a different code page than your client locale, and you want Informix to perform conversions between the code pages.
 - Set the DBNLS environment variable to 1 to ensure that the Informix software verifies that the DB_LOCALE setting matches the actual locale of the Informix database. For example:

```
DBNLS=1
```
3. UNIX: If the **djxlink** script or the **djxlinkInformix** script did not run when you installed WebSphere Federation Server, run the **djxlink** script or the **djxlinkInformix** script.
4. If you changed any environment variables in step 2 or if you ran the **djxlink** script or the **djxlinkInformix** script in step 3, recycle the DB2 instance. When you recycle the DB2 instance, you ensure that the environment variable changes are properly set on the federated server.
 - a. Issue the db2stop command.
 - b. If you ran the **djxlink** script or the **djxlinkInformix** script, issue the db2iupdt command.
 - c. Issue the db2start command.
5. Verify that the migration was successful:
 - a. Connect to the database.
 - b. Issue SELECT, INSERT, UPDATE, and DELETE statements on the nicknames to verify that you can view and modify the data.
 - c. You can also open a pass-through session to the remote database and issue a SELECT statement directly on a remote table.

Accessing Microsoft SQL Server data sources after migrating to WebSphere Federation Server

To access Microsoft SQL Server data sources when you migrate instances and databases or only databases, you might need to set environment variables, run scripts, recycle the DB2 instance, and verify the migration.

About this task

If you migrated only the database, follow all of the steps in this procedure. If you migrated the instance and database on UNIX, you might need to initiate only some of the steps in this procedure. You might need to initiate the following steps:

- Run the `djxlink` script or the `djxlinkMssql` script.
- Recycle the DB2 instance.
- Verify the migration.

If you migrated the instance and database, you might need to initiate these steps if the `djxlink` script or the `djxlinkMssql` script did not run when you installed IBM WebSphere Federation Server.

Procedure

To access Microsoft SQL Server data sources after you migrate to WebSphere Federation Server from WebSphere Information Integrator, version 8.2:

1. Check the settings of the environment variables for Microsoft SQL Server data sources:
 - a. Locate the `db2dj.ini` file that is in `/sqllib/cfg` subdirectory.
 - b. Set the environment variables in the `db2dj.ini` file as necessary. These variables are as follows:

ODBCINI

Full path to the ODBC data source configuration file (`.odbc.ini`). The `.odbc.ini` file contains the list of ODBC data sources that can be accessed, the ODBC driver library to be used for each data source, and connection information for each data source such as hostname/ipaddress and listener port. For example:

```
ODBCINI=/home/db2inst1/.odbc.ini
```

DJX_ODBC_LIBRARY_PATH

The directory path where the ODBC library files reside. For example:

```
DJX_ODBC_LIBRARY_PATH=ODBC_home_directory/lib
```

ODBC_home_directory is the directory path where the ODBC driver is installed.

LD_LIBRARY_PATH

On Solaris, the directory path where the ODBC library files reside. For example:

```
LD_LIBRARY_PATH=ODBC_home_directory/lib
```

SHLIB_PATH

On HP-UX, the directory path where the ODBC library files reside. For example:

```
SHLIB_PATH=ODBC_home_directory/lib
```

2. Check the Microsoft SQL Server configuration:

- UNIX: verify that the /opt/odbc/odbc.ini file is properly set up. Use the demoodbc tool provided with DataDirect Connect to test connectivity to the Microsoft SQL Server data source.
 - Windows: verify that the ODBC System DSN is properly set up. Use a query tool that can access ODBC data sources, or use the connectivity test in the Microsoft SQL Server ODBC driver data source configuration window to test connectivity to the Microsoft SQL Server data source.
3. Verify that the migration was successful:
 - a. Connect to the database.
 - b. Issue SELECT, INSERT, UPDATE, and DELETE statements on the nicknames to verify that you can view and modify the data. You can also open a passthrough session to the remote database and issue a SELECT statement on a remote table.

Related tasks

Configuring access to Microsoft SQL Server data sources

Related reference

Unicode default forward data type mappings - Microsoft SQL Server wrapper

“DataJoiner protocols for Microsoft SQL Server data sources” on page 61

DataJoiner supports several protocols for Microsoft SQL Server data sources.

WebSphere Federation Server supports wrappers for Microsoft SQL Server data sources.

db2iupdt - Update instances command

Accessing ODBC data sources after migrating to WebSphere Federation Server

To access ODBC data sources when you access the data sources on UNIX systems, you set environment variables, issue the ALTER WRAPPER statement to specify the MODULE option, and verify the wrapper option settings for 32-bit and 64-bit clients.

Before you begin

If you use transparent DDL to create remote tables on ODBC data sources through DataJoiner, you might receive errors with SELECT statements after you migrate to WebSphere Federation Server. The DataJoiner DATE data type is mapped to the ODBC DATETIME data type when you create a remote ODBC table through DataJoiner. To avoid these errors, change the local data type from DATE to TIMESTAMP in the federated database systems catalog after you migrate.

About this task

The ODBC wrapper can be used to access a data source if there is not an available data source-specific wrapper, such as NET8 for Oracle data sources, for the data source that you want to access.

Use this procedure only if you are accessing ODBC data sources on UNIX. No post-migration steps are required if you are accessing ODBC data sources on Windows.

Data type mappings that are migrated from DB2 Information Integrator, version 8.1: Some data type mappings for ODBC data sources in IBM WebSphere

Federation Server are not the same as they were in DB2 Information Integrator, version 8.1. The following data type mappings in a Unicode federated databases have changed:

- SQL_WCHAR
- SQL_WVARCHAR
- SQL_WLONGVARCHAR

All of the data type mappings from DB2 Information Integrator, version 8.1 are migrated to WebSphere Federation Server as they are in DB2 Information Integrator, version 8.1. However, if you drop a nickname for the ODBC data source that was created in DB2 Information Integrator, version 8.1. and then recreate that nickname using WebSphere Federation Server, the data type mapping changes to the new type mapping. If you need to use the data type mapping for SQL_WCHAR, SQL_WVARCHAR, or SQL_WLONGVARCHAR from DB2 Information Integrator, version 8.1, you must create forward type mapping or alter the column data type of the nickname.

Procedure

To access ODBC data sources after you migrate to WebSphere Federation Server from WebSphere Information Integrator, version 8.2:

1. If you migrated only the database, set the appropriate environment variables in the `sqllib/cfg/db2dj.ini` file. For example, if you configure a data source by using Data Direct Connect for ODBC, set the following environment variables:

- ODBCINI
- LD_LIBRARY_PATH (Solaris only)

2. Issue the ALTER WRAPPER statement to specify the MODULE option. You must specify the MODULE option to enable the ODBC wrapper on UNIX. For AIX, enclose the shared object member name in parentheses. For example:

```
ALTER WRAPPER ODBC
  OPTIONS (add MODULE '/opt/odbc/lib/libodbc.a(odbc.so)' );
```

In this example, `/opt/odbc/lib/libodbc.a` is the fully qualified path of the directory that contains the ODBC Driver Manager library on AIX. The text `(odbc.so)`, is the appropriate member in the library. The MODULE wrapper option specifies the full path to the ODBC driver manager library.

3. AIX and Solaris: verify the wrapper option settings for 32-bit and 64-bit clients for ODBC data sources.

Option	Description
32-bit client for ODBC data source	<ol style="list-style-type: none"> 1. Query the SYSIBM.SYSWRAPOPTIONS catalog table to verify the DB2_SOURCE_CLIENT_MODE wrapper option is set to 32BIT. If necessary, change the value of the wrapper option by issuing the ALTER WRAPPER statement. The DB2_SOURCE_CLIENT_MODE wrapper option indicates to the federated server that your data source client is 32-bit. 2. Query the SYSIBM.SYSWRAPOPTIONS catalog table to verify that the DB2_FENCED wrapper option is set to Y. If necessary, change the value of the wrapper option by issuing the ALTER WRAPPER statement. The DB2_FENCED wrapper option must be set to Y when the DB2_SOURCE_CLIENT_MODE wrapper option is set.
64-bit client for ODBC data source	<ol style="list-style-type: none"> 1. If the DB2_SOURCE_CLIENT_MODE wrapper option is set, drop the wrapper option by issuing the ALTER WRAPPER statement. 2. If the DB2_FENCED wrapper option was not set before migrating to WebSphere Federation Server, you can drop this wrapper option by using the ALTER WRAPPER statement. 3. If you migrated from WebSphere Information Integrator, version 8.2, and the DB2_FENCED wrapper option was set prior to migration, query the SYSIBM.SYSWRAPOPTIONS catalog table to verify that the DB2_FENCED wrapper option setting was migrated to WebSphere Federation Server.

4. Verify that the migration was successful:
 - a. Connect to the database.
 - b. Issue SELECT, INSERT, UPDATE, and DELETE statements on the nicknames to verify that you can view and modify the data.

You can also open a passthrough session to the remote database and issue a SELECT statement on a remote table.

Related tasks

Configuring access to ODBC data sources

Related reference

“DataJoiner protocols for ODBC data sources” on page 63

DataJoiner supports the Generic protocol for ODBC data sources. WebSphere Federation Server supports the ODBC wrapper for ODBC data sources.

db2look - DB2 statistics and DDL extraction tool command

Unicode default forward data type mappings - ODBC wrapper

Accessing Oracle data sources after migrating to WebSphere Federation Server

Use this procedure to access Oracle data sources if you migrated the instance and database or only the database.

About this task

If you migrated only the database, follow all of the steps in this procedure.

If you migrated the instance and database on UNIX operating systems, you might need to perform only some of the steps in this procedure. The steps that you might need to perform are:

- Running the `djxlink` script or the `djxlinkOracle` script
- Recycling the DB2 instance
- Verifying the migration

If you migrated the instance and database, you might need to perform these steps if the `djxlink` script or the `djxlinkOracle` script did not run when you installed IBM WebSphere Federation Server.

Procedure

To access Oracle data sources using the NET8 wrapper after you migrate to WebSphere Federation Server from DataJoiner or from DB2 Universal Database version 7:

1. Check the settings of the Oracle environment variables:
 - a. Locate the `db2dj.ini` file:
 - On UNIX operating systems, the `db2dj.ini` file is in the `/sqllib/cfg` subdirectory.
 - On Windows operating systems, the `db2dj.ini` file is in the `\sqllib\cfg` subdirectory.
 - b. Set the environment variables in the `db2dj.ini` file as necessary. These variables are:

ORACLE_HOME

The fully qualified directory path where the Oracle client software is installed. For example:

```
ORACLE_HOME=/usr/oracle/8.1.7
```

`/usr/oracle/8.1.7` is the fully qualified path of the Oracle home directory. This environment variable is required.

Optional: ORACLE_BASE

The root of the Oracle client directory tree. For example:

```
ORACLE_BASE=oracle_root_directory
```

Optional: ORA_NLS

The directory where location-specific data is stored for the version of Oracle that you are using. For example:

```
ORACLE_NLS33=oracle_root_directory/ocommon/nls/admin/data
```

In this example, the location-specific data is stored in the `oracle_root_directory/ocommon/nls/admin/data` directory on UNIX federated servers that access Oracle 8.1 data sources.

Optional: TNS_ADMIN

The directory where the tnsnames.ora file is located. Set this parameter only if the location of the tnsnames.ora is different from the default location.

- On UNIX operating systems, the default location for the tnsnames.ora file is the \$ORACLE_HOME/network/admin directory.
- On Windows operating systems, the default location for the tnsnames.ora file is the %ORACLE_HOME%\NETWORK\ADMIN directory.

For example, if the tnsnames.ora file resides in the /home/datajoiner directory, set the variable as follows:

```
TNS_ADMIN=/home/datajoiner
```

2. For UNIX operating systems only: Update the .profile file of the DB2 instance with the Oracle environment variable by adding the following line:

```
export ORACLE_HOME=oracle_home_directory
```

oracle_home_directory is the directory where the Oracle client software is installed.

3. Perform the DB2 instance .profile by entering:

```
. .profile
```

4. If the djxlink script or the djxlinkOracle script did not run when you installed WebSphere Federation Server on UNIX operating systems, run the djxlink script or the djxlinkOracle script.
5. If you changed any environment setting in the preceding steps or if you ran only the djxlink script or the djxlinkOracle script in step 4, recycle the DB2 instance: By recycling the DB2 instance, you ensure that the environment variable changes are properly set on the federated server.
 - a. Issue the db2stop command.
 - b. If you ran the djxlink script or the djxlinkOracle script, issue the db2iupdt command.
 - c. Issue the db2start command.
6. Set up and test the Oracle client configuration files (tnsnames.ora and sqlnet.ora). Use Oracle **sqlplus** to test connectivity to the Oracle server.
7. Verify that the migration was successful:
 - a. Connect to the database.
 - b. Issue SELECT, INSERT, UPDATE, and DELETE statements on the nicknames to verify that you can view and modify the data. You can also open a passthrough session to the remote database, and issue a SELECT statement on a remote table.

Related tasks

Configuring access to Oracle data sources

Related reference

db2iupdt - Update instances command

“DataJoiner protocols for Oracle data sources” on page 58

DataJoiner supports the sqlnet and net8 protocols for Oracle data sources.

WebSphere Federation Server supports only the NET8 wrapper for Oracle data sources.

“Changes to default data type mappings for Oracle NET8” on page 58
Some DataJoiner default data type mappings for Oracle data sources are different from WebSphere Federation Server default data type mappings for Oracle data sources.

Accessing Sybase data sources after migrating to WebSphere Federation Server

Use this procedure to access Sybase data sources if you migrated the instance and database or only the database.

About this task

If you migrated only the database, follow all of the steps in this procedure.

If you migrated the instance and database on UNIX systems, you must do step 1 to restore the interfaces file, and you might need to do additional steps.

If you migrated the instance and database, you must recycle the instance in step 6 and verify the migration in step 7 after you set the SYBASE_OCS environment variable or after you run the djxlink script or the djxlinkSybase script.

Procedure

To access Sybase data sources after you migrate to WebSphere Federation Server from DataJoiner or from DB2 Universal Database version 8:

1. Ensure that the Sybase default interfaces file includes an entry for the server that you specify to connect to the Sybase wrapper. The default file on Windows is %SYBASE%\ini\sql.ini. The default file on UNIX and Linux is \$SYBASE/interface.

Optional: If you do not have write authority on the file or do not want to use the file, you can restore the copy of the interfaces file that you made before migration. You can save this restored file in the \$instancehome/sqllib directory where *instancehome* is the home directory of the instance owner.

2. Check the settings of the environment variables for Sybase data sources:
 - a. Locate the db2dj.ini file:
 - On UNIX systems, the db2dj.ini file is in the /sqllib/cfg subdirectory.
 - On Windows operating systems, the db2dj.ini file is in the \sqllib\cfg subdirectory.
 - b. Set the environment variables in the db2dj.inifile as necessary. These variables are:

SYBASE

The directory path where the Sybase Open Client software is installed. For example:

```
SYBASE=/sybase
```

SYBASE_OCS

You must use the Sybase Open Client, version 12 or later. Set the SYBASE_OCS environment variable to the name of the OCS directory. For example:

```
SYBASE_OCS=OCS-12_5
```

3. UNIX: Update the .profile file of the DB2 instance with the SYBASE and SYBASE_OCS environment variables by issuing the following commands:

```
export SYBASE=sybase_home_directory
export SYBASE_OCS=sybase_ocs_directory
export PATH=$SYBASE/$SYBASE_OCS/bin:$PATH
```

sybase_home_directory is the directory where the Sybase Open Client software is installed. *sybase_ocs_directory* is the name of the OCS directory and is required.

4. UNIX: Initiate the DB2 instance `.profile` by entering: `..profile`
5. UNIX: If the `djxlink` script or the `djxlinkSybase` script did not run when you installed WebSphere Federation Server on UNIX systems, run the `djxlink` script or the `djxlinkSybase` script.
6. If you changed any environment setting in the preceding steps, or if you ran only the `djxlink` script or the `djxlinkSybase` script in step 5, recycle the DB2 instance:
 - a. Issue the `db2stop` command.
 - b. If you ran the `djxlink` script or the `djxlinkSybase` script, issue the `db2iupdt` command.
 - c. Issue the `db2start` command. By recycling the DB2 instance, you ensure that the environment variable changes are properly set on the federated server.
7. Verify that the migration was successful:
 - a. Connect to the database.
 - b. Issue `SELECT`, `INSERT`, `UPDATE`, and `DELETE` statements on the nicknames to verify that you can view and modify the data. You can also open a passthrough session to the remote database, and issue a `SELECT` statement on a remote table.

For information on migrating your DB2 replication environment, see *Migrating to Replication version 9*.

Related reference

“DataJoiner protocols for Sybase data sources” on page 65

DataJoiner supports the `ctlib` and `dblib` protocols for Sybase data sources.

WebSphere Federation Server supports only the `CTLIB` wrapper for Sybase data sources.

`db2iupdt` - Update instances command

Accessing Teradata data sources after migrating to WebSphere Federation Server

Before you can access Teradata data sources on AIX and Solaris, you must check the environment variables, run the `djxlink` script, recycle the DB2 instance, and verify wrapper option settings. On Windows, you must verify that the wrapper option settings are correct.

Before you begin

If you already use the Teradata wrapper, you can upgrade to Teradata release V2R5 by modifying your existing Teradata wrapper configuration or by recreating your entire Teradata wrapper configuration of servers, user mappings, type mappings, function mappings, and nicknames.

About this task

Use this procedure if you migrated the instance and database or only the database.

If you migrated only the database, follow all of the steps in this procedure.

If you migrated the instance and the database, you might need to perform only some of the steps in this procedure. You might need to perform these steps:

- Run the djxlink script or the djxlinkTeradata script.
- Recycle the DB2 instance.
- Verify the migration.

If you migrated the instance and database, you might need to perform these steps if the djxlink script or the djxlinkTeradata script did not run when you installed WebSphere Federation Server.

Procedure

To access Teradata data sources after you migrate to WebSphere Federation Server:

1. Check the settings of the environment variables for the Teradata data sources:
 - a. Locate the db2dj.ini file in the sqllib/cfg subdirectory.
 - b. Set the environment variables in db2dj.ini file as necessary:

COPLIB

The directory in which the Teradata shared library file, libcliv2.so, resides. For example:

```
COPLIB=coplib_directory
```

COPERR

The directory in which the errmsg.cat file resides. For example:

```
COPERR=coperr_directory
```

The libcliv2.so file and the errmsg.cat are usually in the same directory. The default directory is /usr/lib on AIX.

2. If the djxlink script or the djxlinkTeradata script did not run when you installed WebSphere Federation Server, run the djxlink script or the djxlinkTeradata script.
3. If you changed any environment variables in the db2dj.ini file in step 1 or if you ran the djxlink script or the djxlinkTeradata script in step 2, recycle the DB2 instance:
 - a. Issue the db2stop command.
 - b. If you ran the djxlink script or the djxlinkTeradata script, issue the db2iupdt command.
 - c. Issue the db2start command. By recycling the DB2 instance, you ensure that the environment variable changes are properly set on the federated server.
4. AIX, Solaris, and Windows: verify the wrapper option settings:
 - a. Query the SYSIBM.SYSWRAPOPTIONS catalog table to verify the DB2_SOURCE_CLIENT_MODE wrapper option is set to 32BIT. If necessary, change the value of the wrapper option by issuing the ALTER WRAPPER statement. The DB2_SOURCE_CLIENT_MODE wrapper option indicates to the federated server that your data source client is 32-bit.
 - b. Query the SYSIBM.SYSWRAPOPTIONS catalog table to verify the DB2_FENCED wrapper option is set to Y. If necessary, change the value of the wrapper option by issuing the ALTER WRAPPER statement. The DB2_FENCED wrapper option must be set when DB2_SOURCE_CLIENT_MODE wrapper option is set.

5. Use the Teradata BTEQ query tool to test your connection to the Teradata server.
6. Verify that the migration is successful:
 - a. Connect to the database.
 - b. Issue SELECT, INSERT, UPDATE, and DELETE statements on the nicknames to verify that you can view and modify the data. You can also open a passthrough session to the remote database and issue a SELECT statement on a remote table.

Related tasks

Configuring access to Teradata data sources

Related reference

db2iupdt - Update instances command

“DataJoiner protocols for Teradata data sources” on page 67

DataJoiner supports the teradata protocol for Teradata data sources. WebSphere Federation Server supports the Teradata wrapper for Teradata data sources.

Modifying existing Teradata configurations

To upgrade to Teradata release V2R5, you can modify your existing Teradata wrapper.

About this task

If you already use the Teradata wrapper, you can upgrade to Teradata release V2R5 by modifying your existing Teradata wrapper configuration or by recreating your entire Teradata wrapper configuration of servers, user mappings, type mappings, function mappings, and nicknames.

Procedure

To modify your existing Teradata wrapper configuration:

1. Change the Teradata server product version to 2.5 on all of your Teradata servers by using the ALTER SERVER statement:

```
ALTER SERVER MYTERASERVER VERSION 2.5
```

2. Remove the TUPLE_ID_TYPE nickname option on all of your Teradata nicknames by using the ALTER NICKNAME statement: For example:

```
ALTER NICKNAME TERANICKNAME OPTIONS (DROP TUPLE_ID_TYPE)
```

To get a list of all nicknames that need to be changed, use the following SQL statement to return nickname object names, server object names, and server versions for all nicknames that are associated with a TERADATA server type:

```
SELECT T.tabname, S.servername,
       S.serverversion
FROM syscat.servers AS S, syscat.taboptions AS T
WHERE T.option = 'SERVER' AND S.servername = T.setting
AND S.servertype = 'TERADATA'
AND S.serverversion IN ( '2.3', '2.4', '3', '4' )
```

Related tasks

“Recreating Teradata configurations” on page 50

To upgrade to Teradata release V2R5, you can recreate your entire Teradata wrapper configuration of servers, user mappings, type mappings, function mappings, and nicknames.

Altering a nickname

Related reference

ALTER NICKNAME statement

ALTER SERVER statement

Recreating Teradata configurations

To upgrade to Teradata release V2R5, you can recreate your entire Teradata wrapper configuration of servers, user mappings, type mappings, function mappings, and nicknames.

About this task

If you already use the Teradata wrapper, you can upgrade to Teradata release V2R5 by modifying your existing Teradata wrapper configuration or by recreating your entire Teradata wrapper configuration of servers, user mappings, type mappings, function mappings, and nicknames.

Procedure

To recreate your entire Teradata wrapper configuration:

1. Use the db2look command to get the correct syntax for each data source configuration statement. The db2look command generates DDL statements from the metadata in the DB2 catalog.
2. In the db2look command output file, make the following modifications:
 - a. Change all references to the Teradata product version to 2.5 for all server objects.
 - b. For AIX and Solaris: update all CREATE WRAPPER statements for the Teradata wrapper by setting the DB2_FENCED wrapper option and the DB2_SOURCE_CLIENT_MODE wrapper option. For example:

```
CREATE WRAPPER teradata OPTIONS (DB2_FENCED 'Y',  
DB2_SOURCE_CLIENT_MODE '32BIT')
```
3. Drop the Teradata wrapper. When you drop a wrapper, all server definitions, user-defined function mappings, and user-defined data type mappings that are dependent on the wrapper are dropped. All user-defined function mappings, nicknames, user-defined data type mappings, and user mappings that are dependent on the dropped server definitions are also dropped. All packages dependent on the dropped objects and inoperative views are invalidated.
4. If necessary, drop the type and function mappings that are associated with the wrapper that you dropped. The DROP statement drops only the objects that are explicitly or implicitly associated with a wrapper. If other objects are associated with the name of the wrapper that you dropped, you must manually drop those objects.
5. Issue the SQL statements in the db2look output file to reconfigure access to Teradata. For example, if the name of the output file is teradata.sql, issue the following command:

```
db2 -tvf teradata.sql
```
6. Bind any application packages that were invalidated when you dropped the wrapper.

Related tasks

“Modifying existing Teradata configurations” on page 49

To upgrade to Teradata release V2R5, you can modify your existing Teradata wrapper.

Chapter 5. Migration reference

When you migrate to WebSphere Federation Server from DataJoiner or WebSphere Information Integrator, you might see differences in default data type mappings, changes in wrapper library names and the wrappers that support relational and nonrelational data sources, and changes in nonrelational user-defined functions for life sciences.

DataJoiner migration

When you migrate to WebSphere Federation Server some catalog table names and default data type mappings are different from the corresponding catalog table names and default data type mappings in DataJoiner. WebSphere Federation Server provides wrappers to access data sources that are different from the protocols that DataJoiner uses to access data sources.

DataJoiner terms and their federation server equivalents

WebSphere Federation Server uses terms that are different but equivalent to common DataJoiner terms.

The following table lists common DataJoiner terms and their federation server equivalents.

Table 5. DataJoiner terms and their federation server equivalents

DataJoiner term	WebSphere Federation Server term
Data Access Module (DAM)	Wrapper
DataJoiner catalog	Global catalog (sometimes referred to as the federated database system catalog)
DataJoiner database	Federated database
DataJoiner server	Federated server
Two-phase commit	Federated two-phase commit
Protocol	Wrapper
Remote table	Data source object (includes tables, views, synonyms, and other nonrelational objects)
Server mapping	Server definition
Stored procedure nickname	Federated stored procedure

Related concepts

Wrappers and wrapper modules
Server definitions and server options

Related reference

“Migrated objects” on page 6

When instances and databases are migrated to IBM WebSphere Federation Server, objects are migrated. For objects that are not migrated, you might need to perform tasks after you migrate to WebSphere Federation Server or certain database operations might be limited.

DataJoiner catalog conversion

The DataJoiner catalog and the WebSphere Federation Server global catalog each contain information about the federated objects. Some of the corresponding catalog table names are different in WebSphere Federation Server.

The DB2 federated database system catalog is referred to as the *global catalog*.

The following table lists all federated objects in DataJoiner and WebSphere Federation Server and the location of the corresponding table in the DataJoiner catalog and global catalog.

Table 6. Federated objects and catalog information tables in DataJoiner and WebSphere Federation Server

Federated object	DataJoiner catalog table	Global catalog table
Column	syscat.columns	syscat.columns
Column option	Not applicable	syscat.coloptions
Function mappings ¹	syscat.server_functions	syscat.funcmappings
Indexes on nicknames	syscat.indexes	syscat.indexes
Nickname	syscat.tables	syscat.tables and syscat.nicknames
Nickname option	Not applicable	syscat.taboptions
Pass-through privileges	syscat.passthru_auth	syscat.passthruauth
Packages	syscat.packages	syscat.packages
Package dependency	syscat.packagedep	syscat.packagedep
Reverse type mappings ²	syscat.revtypemappings	syscat.typemappings
Server	syscat.servers	syscat.servers
Server option	syscat.server_options	syscat.serveroptions
Stored procedure nickname	syscat.procedures	syscat.routines
Stored procedure nickname parameters	syscat.procparms	syscat.routineparms
Type mappings ²	syscat.server_datatypes	syscat.typemappings
User mappings	syscat.remoteusers	syscat.useroptions
Wrapper	Not applicable	syscat.wrappers

1. In WebSphere Federation Server, default function mappings for a given source type are stored in the wrapper library. In DataJoiner, default function mappings for a given source type are stored in the syscat.server_functions catalog table.

2. In WebSphere Federation Server, forward and reverse type mapping information is available in syscat.typemappings table. The mappingdirection field indicates whether a type mapping is forward "F" or reverse "R".

Not all information that is stored in a single table in the DataJoiner catalog is placed in the corresponding single table in the global catalog. In the global catalog, federated object information can be stored in multiple tables that relate to the same federated object. For example, all server information is stored in the syscat.columns table in the DataJoiner catalog. In the global catalog, all server information is stored in the syscat.columns and syscat.coloptions tables.

The maximum length of the federated object identifiers also differs between DataJoiner and WebSphere Federation Server. The following table lists all federated object identifiers and their maximum lengths in DataJoiner and WebSphere Federation Server.

Table 7. federated object identifier lengths in DataJoiner and WebSphere Federation Server

Identifier for federated object	DataJoiner length	WebSphere Federation Server length
Column name	18	30
Function mapping name	18	128
Index name	18	18
Nickname	18	128
Package name	8	8
Server name	18	128
Schema name	8	128
Stored procedure nickname	8	128
Type mapping name	18	18
Wrapper name	Not applicable	128

Related concepts

Wrappers and wrapper modules
 Server definitions and server options

DB2 data sources

DataJoiner uses multiple protocols to support all DB2 family data sources whereas WebSphere Federation Server uses a wrapper. Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 family data sources.

DataJoiner protocols for DB2 family data sources

DataJoiner uses multiple protocols to support all DB2 family data sources. WebSphere Federation Server uses a single wrapper to support all DB2 family data sources. A wrapper is the mechanism that the federated server uses to communicate with and retrieve data from the data sources.

This topic describes the differences between DataJoiner protocols and the corresponding WebSphere Federation Server wrapper for all DB2 family data sources that are supported in WebSphere Federation Server.

In WebSphere Federation Server, the DRDA wrapper supports the following DB2 family data sources:

- DB2 Database for Linux, UNIX, and Windows
- DB2 Universal Database for z/OS and OS/390
- DB2 Universal Database for iSeries (AS/400®)
- DB2 Server for VSE and VM

In DataJoiner, the following protocols support all DB2 family data sources:

- drda
- drdaIP
- db2ra

In WebSphere Federation Server, the DataJoiner protocols correspond to the single DRDA wrapper. The DRDA wrapper is used for all DB2 family data sources.

Related reference

“Changes to default data type mappings for DB2 Universal Database for z/OS and OS/390 data sources”

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Universal Database for z/OS and OS/390 data sources.

“Changes to default data type mappings for DB2 Universal Database for iSeries (AS/400) data sources” on page 55

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Universal Database for iSeries (AS/400) data sources.

“Changes to default data type mappings for DB2 Database for Linux, UNIX, and Windows data sources” on page 56

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Database for Linux, UNIX, and Windows data sources.

“Changes to default data type mappings for DB2 Server for VSE and VM data sources” on page 57

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Server for VSE and VM data sources.

Changes to default data type mappings for DB2 Universal Database for z/OS and OS/390 data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Universal Database for z/OS and OS/390 data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for DB2 Universal Database for z/OS and OS/390 data sources. The data mappings that remain unchanged between the products are not shown.

Table 8. DB2 Universal Database for z/OS and OS/390 default forward data type mapping changes

DB2 data type	DataJoiner data type	Federated data type
BLOB	-	BLOB
CHAR ¹	LONG VARCHAR	-
CHAR ¹	-	VARCHAR
CLOB	-	CLOB
DBCLOB	-	DBCLOB
LONGVAR	LONG VARCHAR	CLOB
LONGVARG	LONG VARGRAPHIC	DBCLOB
VARCHAR ¹	LONG VARCHAR	-
VARCHAR ¹	-	VARCHAR
VARGRAPHIC	LONG VARGRAPHIC	-
VARGRAPHIC	-	VARGRAPHIC
1. The upper and lower lengths are different for these two data type mappings.		

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for DB2 for z/OS and OS/390 data sources. The data mappings that remain unchanged between the products are not shown.

Table 9. DB2 for z/OS and OS/390 default reverse data type mapping changes

Federated data type	DB2 data type used in DataJoiner	DB2 data type used in federated systems
BLOB	LONGVAR	BLOB
CLOB	LONGVAR	CLOB
DBCLOB	-	DBCLOB
LONG VARCHAR	LONG VAR	-
LONG VARGRAPHIC	LONG VARGRAPHIC	-

Related reference

“DataJoiner protocols for DB2 family data sources” on page 53

DataJoiner uses multiple protocols to support all DB2 family data sources. WebSphere Federation Server uses a single wrapper to support all DB2 family data sources. A wrapper is the mechanism that the federated server uses to communicate with and retrieve data from the data sources.

Default forward data type mappings

Default reverse data type mappings

DB2 for z/OS data sources

Changes to default data type mappings for DB2 Universal Database for iSeries (AS/400) data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Universal Database for iSeries (AS/400) data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for DB2 Universal Database for iSeries (AS/400) data sources. The data mappings that remain unchanged between the products are not shown.

Table 10. DB2 Universal Database for iSeries (AS/400) default forward data type mapping changes

DB2 data type	DataJoiner data type	Federated data type
BLOB	-	BLOB
CHAR	LONG VARCHAR	-
CHAR	-	VARCHAR
CLOB	-	CLOB
DBCLOB	-	DBCLOB
LONGVARCHAR	-	CLOB
LONGVARGRAPHIC	LONG VARGRAPHIC	DBCLOB
VARCHAR	LONG VARCHAR	-
VARCHAR	-	VARCHAR
VARGRAPHIC	LONG VARGRAPHIC	-
VARGRAPHIC	-	VARGRAPHIC

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for DB2 Universal Database for iSeries (AS/400) data sources. The data mappings that remain unchanged between the products are not shown.

Table 11. DB2 Universal Database for iSeries (AS/400) default reverse data type mapping changes

Federated data type	DB2 data type used in DataJoiner	DB2 data type used in federated systems
LONG VARCHAR	LONG VARCHAR	CLOB
LONG VARGRAPHIC	LONG VARGRAPHIC	DBCLOB

Related reference

“DataJoiner protocols for DB2 family data sources” on page 53

DataJoiner uses multiple protocols to support all DB2 family data sources. WebSphere Federation Server uses a single wrapper to support all DB2 family data sources. A wrapper is the mechanism that the federated server uses to communicate with and retrieve data from the data sources.

Default forward data type mappings

Default reverse data type mappings

Changes to default data type mappings for DB2 Database for Linux, UNIX, and Windows data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Database for Linux, UNIX, and Windows data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for DB2 Database for Linux, UNIX, and Windows data sources. The data mappings that remain unchanged between the product are not shown.

Table 12. DB2 Database for Linux, UNIX, and Windows default forward data type mapping changes

DB2 data type	DataJoiner data type	Federated data type
BLOB	-	BLOB
CHAR	LONG VARCHAR	-
CHAR	VARCHAR	-
CHAR	-	CHAR
CLOB	-	CLOB
DBCLOB	-	DBCLOB
LONG VARCHAR	LONG VARCHAR	CLOB
LONG VARGRAPHIC	LONG VARGRAPHIC	DBCLOB
VARCHAR	LONG VARCHAR	-
VARCHAR	-	VARCHAR
VARGRAPHIC	LONG VARGRAPHIC	-
VARGRAPHIC	-	VARGRAPHIC

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for DB2 Database for

UNIX and Windows data sources. The data mappings that remain unchanged between the product are not shown.

Table 13. DB2 Database for Linux, UNIX, and Windows default reverse data type mapping changes

Federated data type	DB2 data type used in DataJoiner	DB2 data type used in federated systems
LONG VARCHAR	LONG VARCHAR	CLOB
LONG VARGRAPHIC	LONG VARGRAPHIC	DBCLOB

Related reference

“DataJoiner protocols for DB2 family data sources” on page 53

DataJoiner uses multiple protocols to support all DB2 family data sources. WebSphere Federation Server uses a single wrapper to support all DB2 family data sources. A wrapper is the mechanism that the federated server uses to communicate with and retrieve data from the data sources.

Default forward data type mappings

Default reverse data type mappings

Changes to default data type mappings for DB2 Server for VSE and VM data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for DB2 Server for VSE and VM data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for DB2 Server for VSE and VM data sources. The data mappings that remain unchanged between the product are not shown.

Table 14. DB2 Server for VSE and VM default forward data type mappings

DB2 data type	DataJoiner data type	Federated data type
BLOB	-	BLOB
CLOB	-	CLOB
DBCLOB	-	DBCLOB
CHAR	VARCHAR	-
CHAR	LONG VARCHAR	-
VARCHAR	LONG VARCHAR	-
VARCHAR	-	VARCHAR
LONGVARCHAR	LONG VARCHAR	CLOB
VARGRAPHIC	LONG VARGRAPHIC	-
LONGVARGRAPHIC	LONG VARGRAPHIC	DBCLOB
VARGRAPHIC	-	VARGRAPHIC

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for DB2 Server for VSE and VM data sources. The data mappings that remain unchanged between the product are not shown.

Table 15. DB2 Server for VSE and VM default reverse data type mappings

Federated data type	DB2 data type used in DataJoiner	DB2 data type used in federated systems
BLOB	-	BLOB
CLOB	-	CLOB
DBCLOB	-	DBCLOB
LONG VARCHAR	LONG VARCHAR	-
VARGRAPHIC	LONG VARGRAPHIC	-
VARGRAPHIC	-	-

Related reference

“DataJoiner protocols for DB2 family data sources” on page 53

DataJoiner uses multiple protocols to support all DB2 family data sources. WebSphere Federation Server uses a single wrapper to support all DB2 family data sources. A wrapper is the mechanism that the federated server uses to communicate with and retrieve data from the data sources.

Default forward data type mappings

Default reverse data type mappings

Oracle data sources

DataJoiner uses protocols to support Oracle data sources whereas WebSphere Federation Server uses a wrapper. Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Oracle data sources.

DataJoiner protocols for Oracle data sources

DataJoiner supports the sqlnet and net8 protocols for Oracle data sources.

WebSphere Federation Server supports only the NET8 wrapper for Oracle data sources.

If you use the sqlnet or the net8 protocols in DataJoiner, after you migrate to WebSphere Federation Server, you must use the NET8 wrapper to access Oracle data sources.

Related tasks

“Accessing Oracle data sources after migrating to WebSphere Federation Server” on page 44

Use this procedure to access Oracle data sources if you migrated the instance and database or only the database.

Related reference

“Changes to default data type mappings for Oracle NET8”

Some DataJoiner default data type mappings for Oracle data sources are different from WebSphere Federation Server default data type mappings for Oracle data sources.

Changes to default data type mappings for Oracle NET8

Some DataJoiner default data type mappings for Oracle data sources are different from WebSphere Federation Server default data type mappings for Oracle data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for the Oracle NET8

wrapper. The data mappings that remain unchanged between the product are not shown.

Table 16. Oracle NET8 default forward data type mapping changes

Oracle data type	DataJoiner data type	Federated data type
BLOB	-	BLOB
CLOB	-	CLOB
LONG	CLOB	CLOB
LONG RAW	BLOB	BLOB
NCHAR(1-127) ²	-	GRAPHIC
NCHAR(128-2000) ²	-	VARGRAPHIC
NCLOB ²	-	DBCLOB
NVARCHAR ²	-	VARGRAPHIC
TIMESTAMP ¹	-	TIMESTAMP
1. Requires Oracle client and server version 9 or later.		
2. Requires Oracle client and server version 9 or later and local UNICODE codepage.		

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for the Oracle NET8 wrapper. The data mappings that remain unchanged between the product are not shown.

Table 17. Oracle NET8 default reverse data type mapping changes

Federated data type	Oracle data type used in DataJoiner	Oracle data type used in federated systems
BLOB	LONG RAW	BLOB
CLOB	LONG	CLOB
DBCLOB ²	-	NCLOB
FLOAT	-	FLOAT
GRAPHIC ²	-	NCHAR
LONG VARCHAR	LONG	-
LONG VARCHAR	LONG RAW	-
REAL	-	FLOAT (length: 63)
VARCHAR ¹	LONG RAW	RAW
VARCHAR ¹	LONG RAW	-
VARCHAR	LONG	VARCHAR2
VARGRAPHIC ²	-	NVARCHAR2
1. The upper and lower lengths are different for these two data type mappings.		
2. Requires Oracle client and server version 9 or later and local UNICODE codepage.		

Related tasks

“Accessing Oracle data sources after migrating to WebSphere Federation Server” on page 44

Use this procedure to access Oracle data sources if you migrated the instance and database or only the database.

Related reference

“DataJoiner protocols for Oracle data sources” on page 58

DataJoiner supports the sqlnet and net8 protocols for Oracle data sources. WebSphere Federation Server supports only the NET8 wrapper for Oracle data sources.

Default forward data type mappings

Default reverse data type mappings

Informix data sources

DataJoiner uses protocols to support Informix data sources whereas WebSphere Federation Server uses a wrapper. Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Informix data sources.

DataJoiner protocols for Informix data sources

DataJoiner protocols have a corresponding WebSphere Federation Server wrapper for Informix data sources.

The following table lists the Informix data sources that are supported in WebSphere Federation Server, the DataJoiner protocol names for each data source, and the corresponding WebSphere Federation Server wrapper name.

Table 18. Informix data source protocol names and their corresponding wrapper names

Data source	DataJoiner protocol name	WebSphere Federation Server wrapper name
Informix version 7	informix7	INFORMIX
Informix version 9	informix9	INFORMIX

Related reference

“Changes to default data type mappings for Informix data sources”

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Informix data sources.

Changes to default data type mappings for Informix data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Informix data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server default forward data type mappings for all Informix data sources. The data mappings that remain unchanged between the product are not shown.

Table 19. Informix default forward data type mapping changes

Informix data type	DataJoiner data type	Federated data type
CHAR	LONG VARCHAR	VARCHAR
NCHAR	LONG VARCHAR	VARCHAR
SMALLFLOAT	DOUBLE	REAL

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for all Informix data sources. The data mappings that remain unchanged between the product are not shown.

Table 20. Informix default reverse data type mapping changes

Federated data type	Informix data type used in DataJoiner	Informix data type used in federated systems
VARCHAR	TEXT(This type mapping has a length that ranges from 255 to 4000.)	TEXT(This type mapping is only valid with Informix version 7 and version 8, and has a length that ranges from 255 to 32672.)
VARCHAR	-	LVARCHAR(This type mapping is only valid with Informix version 9 and version 10 and has a length that ranges from 255 to 2048.)
VARCHAR	-	TEXT(This type mapping is only valid with Informix version 8, version 9, and version 10 and has a length that ranges from 2049 to 32672.)

Related reference

“DataJoiner protocols for Informix data sources” on page 60

DataJoiner protocols have a corresponding WebSphere Federation Server wrapper for Informix data sources.

Default forward data type mappings

Default reverse data type mappings

Microsoft SQL data sources

DataJoiner uses protocols to support Microsoft SQL data sources whereas WebSphere Federation Server uses a wrapper. Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Microsoft SQL data sources.

DataJoiner protocols for Microsoft SQL Server data sources

DataJoiner supports several protocols for Microsoft SQL Server data sources. WebSphere Federation Server supports wrappers for Microsoft SQL Server data sources.

The following table lists the DataJoiner protocols for accessing Microsoft SQL Server data sources and the corresponding WebSphere Federation Server wrapper names.

Table 21. Microsoft SQL Server data source protocol names and their corresponding wrapper names

Data source	DataJoiner protocol name	Default WebSphere Federation Server wrapper name
Microsoft SQL Server on UNIX	<ul style="list-style-type: none"> For AIX: mssqlodbc and djxsqlnk For the Solaris Operating Environment: mssqlodbc or djxsqlnk 	MSSQLODBC3

Table 21. Microsoft SQL Server data source protocol names and their corresponding wrapper names (continued)

Data source	DataJoiner protocol name	Default WebSphere Federation Server wrapper name
Microsoft SQL Server on Windows	djxmssql	DJXMSSQL3

Related tasks

“Accessing Microsoft SQL Server data sources after migrating to WebSphere Federation Server” on page 40

To access Microsoft SQL Server data sources when you migrate instances and databases or only databases, you might need to set environment variables, run scripts, recycle the DB2 instance, and verify the migration.

Related reference

Default forward data type mappings

Default reverse data type mappings

“Changes to default data type mappings for Microsoft SQL Server data sources”

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Microsoft SQL Server data sources.

“DataJoiner protocols for ODBC data sources” on page 63

DataJoiner supports the Generic protocol for ODBC data sources. WebSphere Federation Server supports the ODBC wrapper for ODBC data sources.

Changes to default data type mappings for Microsoft SQL Server data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Microsoft SQL Server data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for all Microsoft SQL Server data sources. In data types where the same remote data type maps to different DB2 data types, the mapping that is used depends on the data type length. The data mappings that remain unchanged between the product are not shown.

Table 22. Microsoft SQL Server default forward data type mapping changes

Microsoft SQL Server data type	DataJoiner data type	Federated data type
binary	LONG VARCHAR	VARCHAR
char	LONG VARCHAR	VARCHAR
float	DOUBLE	REAL
floatn	DOUBLE	REAL
nchar	CHARACTER	CHARACTER
nchar	CHARACTER	VARCHAR
nchar	VARCHAR	VARCHAR
nchar	LONG VARCHAR	-

Table 22. Microsoft SQL Server default forward data type mapping changes (continued)

Microsoft SQL Server data type	DataJoiner data type	Federated data type
nvarchar	VARCHAR	VARCHAR
nvarchar	LONG VARCHAR	-
real	DOUBLE	REAL
varbinary	LONG VARCHAR	VARCHAR
varchar	LONG VARCHAR	VARCHAR

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for all Microsoft SQL Server data sources. The data mappings that remain unchanged between the two products are not shown.

Table 23. Microsoft SQL Server default reverse data type mapping changes

Federated data type	Microsoft SQL server data type used in DataJoiner	Microsoft SQL server data type used in federated systems
LONG VARCHAR	text	-
VARCHAR	char	varchar
VARCHAR	text	varchar
VARCHAR	-	varchar
VARCHAR	-	text
VARCHAR FOR BIT DATA	image	varbinary
VARCHAR FOR BIT DATA	-	varbinary
VARCHAR FOR BIT DATA	-	image

Related reference

“DataJoiner protocols for Microsoft SQL Server data sources” on page 61
 DataJoiner supports several protocols for Microsoft SQL Server data sources.
 WebSphere Federation Server supports wrappers for Microsoft SQL Server data sources.

Default forward data type mappings

Default reverse data type mappings

ODBC data sources

DataJoiner uses a protocol to support ODBC data sources whereas WebSphere Federation Server uses a wrapper. Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for ODBC data sources.

DataJoiner protocols for ODBC data sources

DataJoiner supports the Generic protocol for ODBC data sources. WebSphere Federation Server supports the ODBC wrapper for ODBC data sources.

Any data source that can use ODBC driver 3.0 or later is a potential ODBC data source. For example, Microsoft SQL Server can use the ODBC driver.

Related tasks

“Accessing ODBC data sources after migrating to WebSphere Federation Server” on page 41

To access ODBC data sources when you access the data sources on UNIX systems, you set environment variables, issue the ALTER WRAPPER statement to specify the MODULE option, and verify the wrapper option settings for 32-bit and 64-bit clients.

Related reference

“DataJoiner protocols for Microsoft SQL Server data sources” on page 61

DataJoiner supports several protocols for Microsoft SQL Server data sources. WebSphere Federation Server supports wrappers for Microsoft SQL Server data sources.

“Changes to default data type mappings for ODBC data sources”

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for ODBC data sources.

Changes to default data type mappings for ODBC data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for ODBC data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for ODBC data sources. The data mappings that remain unchanged between the products are not shown.

Table 24. ODBC forward default data type mapping changes

ODBC data type	DataJoiner data type	Federated data type
SQL_BIGINT	DECIMAL	BIGINT
SQL_CHAR (1-254)	LONG VARCHAR	CHAR
SQL_CHAR (255-32672)	LONG VARCHAR	VARCHAR
SQL_BINARY	LONG VARCHAR	CHAR FOR BIT DATA
SQL_BINARY	LONG VARCHAR	VARCHAR FOR BIT DATA
SQL_VARCHAR	LONG VARCHAR	VARCHAR
SQL_VARBINARY	LONG VARCHAR	VARCHAR FOR BIT DATA
SQL_WCHAR (1-127)	-	CHAR
SQL_WCHAR (128-16336)	-	VARCHAR
SQL_WVARCHAR	-	VARCHAR
SQL_WLONGVARCHAR	-	CLOB

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for ODBC data sources. The data mappings that remain unchanged between the product are not shown.

Table 25. ODBC reverse default data type mapping changes

Federated data type	ODBC data type used in DataJoiner	ODBC data type used in Federated systems
BLOB	-	SQL_LONGVARBINARY
BIGINT	-	SQL_BIGINT
CHAR	-	SQL_CHAR

Table 25. ODBC reverse default data type mapping changes (continued)

Federated data type	ODBC data type used in DataJoiner	ODBC data type used in Federated systems
CHAR FOR BIT DATA	-	SQL_BINARY
CLOB	-	SQL_LONGVARCHAR
DATE	-	SQL_TYPE_DATE
DECIMAL	-	SQL_DECIMAL
DOUBLE	-	SQL_DOUBLE
FLOAT	-	SQL_FLOAT
INTEGER	-	SQL_INTEGER
NUMERIC	-	SQL_DECIMAL
REAL	-	SQL_REAL
SMALLINT	-	SQL_SMALLINT
TIME	-	SQL_TYPE_TIME
TIMESTAMP	-	SQL_TYPE_TIMESTAMP
VARCHAR	-	SQL_VARCHAR
VARCHAR FOR BIT DATA	-	SQL_VARBINARY

Related reference

“DataJoiner protocols for ODBC data sources” on page 63
 DataJoiner supports the Generic protocol for ODBC data sources. WebSphere Federation Server supports the ODBC wrapper for ODBC data sources.
 Default forward data type mappings
 Default reverse data type mappings

Sybase data sources

DataJoiner uses protocols to support Sybase data sources whereas WebSphere Federation Server uses a wrapper. Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Sybase data sources.

DataJoiner protocols for Sybase data sources

DataJoiner supports the ctdlib and dtdlib protocols for Sybase data sources. WebSphere Federation Server supports only the CTLIB wrapper for Sybase data sources.

If you use the dtdlib protocol in DataJoiner, after you migrate to WebSphere Federation Server, you must use the CTLIB wrapper to access Sybase data sources.

Related tasks

“Accessing Sybase data sources after migrating to WebSphere Federation Server” on page 46
 Use this procedure to access Sybase data sources if you migrated the instance and database or only the database.

Related reference

“Changes to default data type mappings for Sybase data sources” on page 66
 Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Sybase data sources.

Changes to default data type mappings for Sybase data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Sybase data sources.

The following table lists the differences between DataJoiner, WebSphere Information Integrator and WebSphere Federation Server in the default forward data type mappings for Sybase data sources. The data mappings that remain unchanged between the product are not shown.

Table 26. Sybase default forward data type mapping changes

Sybase data type	DataJoiner data type	WebSphere Information Integrator data type	WebSphere Federation Server data type
BINARY	LONG VARCHAR	VARCHAR	CHAR FOR BIT DATA
CHAR	LONG VARCHAR	VARCHAR	CHAR
NCHAR	LONG VARCHAR	VARCHAR	CHAR
NVARCHAR	LONG VARCHAR	VARCHAR	VARCHAR
REAL	DOUBLE	REAL	REAL
VARBINARY	LONG VARCHAR	VARCHAR	VARCHAR FOR BIT DATA
VARCHAR	LONG VARCHAR	VARCHAR	VARCHAR

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for Sybase data sources. The data mappings that remain unchanged between the product are not shown.

Table 27. Sybase default reverse data type mapping changes

Federated data type	Sybase data type used in DataJoiner	Sybase data type used in federated systems
VARCHAR	CHAR	VARCHAR
VARCHAR FOR BIT DATA	BINARY	VARBINARY
LONG VARCHAR	TEXT	-
LONG VARCHAR	IMAGE	-

Related reference

“DataJoiner protocols for Sybase data sources” on page 65

DataJoiner supports the ctlib and dblink protocols for Sybase data sources.

WebSphere Federation Server supports only the CTLIB wrapper for Sybase data sources.

Default forward data type mappings

Default reverse data type mappings

Teradata data sources

DataJoiner uses a protocol to support Teradata data sources whereas WebSphere Federation Server uses a wrapper. Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Teradata data sources.

DataJoiner protocols for Teradata data sources

DataJoiner supports the teradata protocol for Teradata data sources. WebSphere Federation Server supports the Teradata wrapper for Teradata data sources.

Related tasks

“Accessing Teradata data sources after migrating to WebSphere Federation Server” on page 47

Before you can access Teradata data sources on AIX and Solaris, you must check the environment variables, run the djxlink script, recycle the DB2 instance, and verify wrapper option settings. On Windows, you must verify that the wrapper option settings are correct.

Related reference

“Changes to default data type mappings for Teradata data sources”

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Teradata data sources.

Changes to default data type mappings for Teradata data sources

Some DataJoiner default data type mappings are different from WebSphere Federation Server default data type mappings for Teradata data sources.

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default forward data type mappings for the Teradata data sources. The data mappings that remain unchanged between the products are not shown.

Table 28. Teradata default forward data type mapping changes

Teradata data type	DataJoiner data type	Federated data type
BYTE	VARCHAR / LONG VARCHAR FOR BIT DATA	VARCHAR FOR BIT DATA / CHAR FOR BIT DATA / BLOB
CHAR	VARCHAR / LONG VARCHAR	VARCHAR / CHAR / CLOB
VARBYTE	VARCHAR FOR BIT DATA / LONG VARCHAR FOR BIT DATA	VARCHAR FOR BIT DATA / BLOB
VARCHAR	VARCHAR / LONG VARCHAR	VARCHAR / CLOB
VARGRAPHIC	VARGRAPHIC / LONG VARGRAPHIC	VARGRAPHIC / DBCLOB

The following table lists the differences between DataJoiner and WebSphere Federation Server in the default reverse data type mappings for the Teradata data sources. The data mappings that remain unchanged between the products are not shown.

Table 29. Teradata default reverse data type mapping changes

Federated type name	Teradata data type used in DataJoiner	Teradata data type used in federated systems
BLOB	-	VARBYTE
CLOB	-	VARCHAR
DBCLOB	-	VARGRAPHIC

Table 29. Teradata default reverse data type mapping changes (continued)

Federated type name	Teradata data type used in DataJoiner	Teradata data type used in federated systems
LONG VARCHAR	VARCHAR	-
LONG VARCHAR FOR BIT DATA	VARBYTE	-
LONG VARGRAPHIC	VARGRAPHIC	-

Related reference

“DataJoiner protocols for Teradata data sources” on page 67

DataJoiner supports the teradata protocol for Teradata data sources. WebSphere Federation Server supports the Teradata wrapper for Teradata data sources.

Default forward data type mappings

Default reverse data type mappings

Nonrelational data sources for life sciences

Support for KEGG API, version 3.2 in WebSphere Information Integrator, version 8.1 fix pack 9 and later introduced changes to the KEGG user-defined functions. Because of these changes to the KEGG function arguments and function names, you might need to update federated databases that were migrated to WebSphere Federation Server from earlier versions of WebSphere Information Integrator. WebSphere Information Integrator, version 8.1 fix pack 9 included changes to fixed input and output columns for BLAST nicknames. Because of these changes to the input and output columns, you might need to drop existing BLAST nicknames from federated databases that you migrated and create the BLAST nicknames in WebSphere Federation Server.

Changes to fixed columns for BLAST nicknames

New fixed input and output columns for BLAST nicknames are available when you install the latest fix pack of WebSphere Information Integrator. You can specify BLAST input columns in a WHERE clause to pass standard BLAST switches to BLAST. To use the new input and output columns, you must drop existing BLAST nicknames and create the nicknames again.

Table 30. New fixed input columns for BLAST nicknames

Name	Data type	Operators	Description
Mask_Lower_Case	CHAR(1)	=	Use lowercase filtering with a FASTA sequence
Query_Genetic_Code	INTEGER	=	Query genetic code uses default = 1.
DB_Genetic_Code	INTEGER	=	DB genetic code for tblastn and tblastx queries uses default = 1.
XDropoff_Ungapped	DOUBLE	=	The X dropoff value for ungapped extension measured in bits. The value 0.0 invokes the default behavior. For blastn queries, the default is 20 bits. For megablast queries, the default is 10 bits. For all other query types, the default is 7 bits (REAL data types).

Table 30. New fixed input columns for BLAST nicknames (continued)

Name	Data type	Operators	Description
XDropoff_Gapped	INTEGER	=	The X dropoff value for gapped alignment, measured in bits. The value 0 invokes the default behavior. For blastn queries, the default is 30 bits. For megablast queries, the default is 20 bits. For all other query types, the default is 15 bits (INTEGER data types).
Final_XDropoff	INTEGER	=	The X dropoff value for the final gapped alignment, measured in bits. The value 0 invokes the default behavior. For blastn and megablast queries, the default is 50 bits. For tblastx queries, the default is 0 bits. For all other query types, the default is 25 bits (INTEGER data types).

There are new BLAST switches for the new input columns.

Table 31. BLAST search types and switches supported by the input fixed columns

Name	BLAST search types	BLAST switch	Required	Default
Mask_Lower_Case	n, p, x, tn, tx	-U	No	F
Query_Genetic_Code	n, p, x, tn, tx	-Q	No	1
DB_Genetic_Code	tn, tx	-D	No	1
XDropoff_Ungapped	n, p, x, tn, tx	-y	No	0.0
XDropoff_Gapped	n, p, x, tn, tx	-X	No	0
Final_XDropoff	n, p, x, tn, tx	-Z	No	0

There is one new fixed output column for BLAST nicknames.

Table 32. New fixed output columns for BLAST nicknames

Name	Data type	Description
Hsp_Score	DOUBLE	The computed score for an HSP (high-scoring segment pairs) as reported in the BLAST results.

Related tasks

Dropping a nickname

Registering nicknames for BLAST data sources

Related reference

DROP statement

CREATE NICKNAME statement

CREATE NICKNAME statement syntax - BLAST wrapper

CREATE NICKNAME statement - Examples for BLAST wrapper

Changes to KEGG function arguments

KEGG function arguments were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9 and later to support the version 3.2 KEGG API.

KEGG user-defined functions were updated to include new arguments, were renamed, or were removed.

The following table describes the changes to the arguments that are used with the KEGG user-defined functions.

Table 33. KEGG UDF argument changes

Former argument name	New argument name	Description
cpdlist	compound_id	The unique identifier for a compound. The format that you use for the <code>compound_id</code> argument is <code>cpd:compound</code> . The abbreviation for the compound database is <code>cpd</code> . <code>compound</code> is the compound number that is in the KEGG LIGAND composite database. For example, the compound identifier for methylparabanic acid is <code>cpd:C11116</code> .
	compound_id_list	This argument is new in Fix Pack 9. A space-delimited list of compound identifiers.
enzymelist	enzyme_id	The unique identifier for an enzyme. The format that you use for the <code>enzyme_id</code> argument is <code>ec:enzyme</code> . The abbreviation for the enzyme database is <code>ec</code> . <code>enzyme</code> is the enzyme code that is in the enzyme database. For example, the compound identifier for sulfite oxidase is <code>ec:1.8.3.1</code> .
	enzyme_id_list	This argument is new in Fix Pack 9. A space-delimited list of enzyme identifiers.
keggid	genes_id	A unique identifier for a gene. The format that you use for the <code>genes_id</code> argument is <code>org:gene_name</code> . The three-letter KEGG code for an organism is <code>org</code> . <code>gene_name</code> is the name of the gene. For example, the genes identifier for the E. coli gene is <code>eco:b0001</code> .
	gene_id_list	This argument is new in Fix Pack 9. A space-delimited list of enzyme identifiers.
	max_result	This argument is new in Fix Pack 9. An integer that is used to specify the number of results that are returned. The <code>max_result</code> argument is used with the <code>start</code> argument to control the results that are returned. For example, if 5 is specified for <code>start</code> and 12 is specified for <code>max_result</code> , the results that are returned start with result 5 and end with result 16.

Table 33. KEGG UDF argument changes (continued)

Former argument name	New argument name	Description
midlist	motif_id	<p>A unique identifier for a motif. The format that you use for the <code>motif_id</code> argument is <code>motif_database_identifier:motif_name</code>. <code>motif_database_identifier</code> is the two-letter code for a motif database. <code>motif_name</code> is the name of the motif. The valid motif database identifiers are:</p> <ul style="list-style-type: none"> • The <code>b1</code> Identifier for the Blocks Motif database • The <code>pd</code> Identifier for the Prodom Motif database • The <code>pf</code> Identifier for the Pfam Motif database • The <code>pr</code> Identifier for the Prints Motif database • The <code>ps</code> Identifier for the Prosite Motif database <p>For example, the motif identifier for the <code>DnaJ</code> entry in the <code>pfam</code> database is <code>pf:DnaJ</code>.</p>
orglist	org	<p>A three-letter KEGG code for an organism. If this argument is not specified, the entire organism list is searched.</p> <p>Each organism in the KEGG databases is assigned a code. The list of organisms changes frequently. Check the current list of genomes for the correct codes at www.genome.ad.jp/kegg/kegg2.html#genes.</p>
pathwayid	pathway_id	<p>A unique identifier for a pathway. The format that you use for the <code>pathway_id</code> argument is <code>path:pathway_number</code>. <code>path</code> is the abbreviation for the pathway database. <code>pathway_number</code> consists of a prefix and a number. The valid prefixes are <code>map</code> and <code>org</code>. The <code>map</code> prefix indicates that you want to search for the reference pathway. The <code>org</code> prefix is a three-letter KEGG code for an organism.</p> <p>For example, the pathway identifier for the reference pathway for the cytrate cycles is <code>path:map00020</code></p> <p>For example, the pathway identifier for the specific pathways for the E.coli genes is <code>path:eco00020</code>.</p>
	start	<p>This argument is new in Fix Pack 9.</p> <p>An integer that is used to specify the first result that is returned. The <code>start</code> argument is used with the <code>max_results</code> argument to control the results that are returned. For example, if 5 is specified for <code>start</code> and 12 is specified for <code>max_results</code>, the results that are returned start with result 5 and end with result 16.</p>

Table 33. KEGG UDF argument changes (continued)

Former argument name	New argument name	Description
threshold		This argument is no longer supported.

Related reference

“Changes to the SSDB function names and accepted arguments”
KEGG function arguments for the SSDB database were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9 to support the version 3.2 KEGG API.

“Changes to SSDB database queries” on page 73
The columns that are returned from SSDB database queries were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9. The columns that are returned depend on the function that you use and the type of row that is returned.

Changes to the SSDB function names and accepted arguments

KEGG function arguments for the SSDB database were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9 to support the version 3.2 KEGG API.

Some function names and function arguments were changed for KEGG user-defined functions for the SSDB database.

Table 34. Changes to the SSDB functions and arguments

Former function name	New function name	New function arguments	New function description
AllNbrsByGene	GetNbrsByGene	genes_id, org, start, max_results	Use the GetNbrsByGeneS and GetNbrsByGeneT functions to search for organisms that are homologous neighbors to the gene that you specify.
BestNbrsByGene	No name change	genes_id, start, max_results	Use the BestNbrsByGeneS and BestNbrsByGeneT functions to search for the best neighbors or the best homologous neighbors of a gene in all organisms.
BstBstNbrsByGene	No name change	genes_id, start, max_results	Use the BstBstNbrsByGeneS and BstBstNbrsByGeneT functions to search for the best-best neighbors or the best-best homologous neighbors of a gene in all organisms.

Table 34. Changes to the SSDB functions and arguments (continued)

Former function name	New function name	New function arguments	New function description
GenesByMotifsT	No name change	motif_id_list, start, max_results	Use the GenesByMotifsT function to search for all of the genes that contain all of the motifs in a list of motifs that you specify. This function now returns a Definition row. In version 2.3, this function returned the keggid and definition.
ParalogsByGene	No name change	genes_id, start, max_results	Use the ParalogsByGeneS and ParalogsByGeneT functions to search for the genes that are paralogous to the gene that you specify.
RevBstNbrsByGene	No name change	genes_id, start, max_results	Use the RevBestNbrsByGeneS and RevBestNbrsByGeneT functions to search for the reverse best neighbors of a gene in all organisms.
ScoreBtwGenes	SimilarBtwGenesT	genes_id, genes_id2	Use the SimilarBtwGenesT function to determine the Smith-Waterman score and the alignment positions between two genes. This function now returns an SSDB row. In version 2.3, this function returned a DOUBLE data type.

Related reference

“Changes to KEGG function arguments” on page 69

KEGG function arguments were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9 and later to support the version 3.2 KEGG API.

“Changes to SSDB database queries”

The columns that are returned from SSDB database queries were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9. The columns that are returned depend on the function that you use and the type of row that is returned.

Changes to SSDB database queries

The columns that are returned from SSDB database queries were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9. The columns that are returned depend on the function that you use and the type of row that is returned.

The table functions in the version 2.3 KEGG user-defined functions returned SSDB rows.

In the version 3.2 KEGG user-defined functions, the following types of rows are returned:

- SSDB rows
- Motif rows
- Definition rows

Changes to the SSDB rows

The columns that are returned with the SSDB rows were changed. The following table shows the changes to the columns.

Table 35. SSDB rows from the SSDB database

version 2.3 column name	version 3.2 column name	Column data type	Description
keggid1	genes_id1	VARCHAR(100)	The identifier for the gene that is specified in the query.
keggid2	genes_id2	VARCHAR(100)	The identifier for the gene that is returned from the query.
sw_score	sw_score	INTEGER	The Smith-Waterman score between genes_id1 and genes_id2.
N/A	bit_score	DOUBLE	The bit score between genes_id1 and genes_id2
identity	identity	DOUBLE	The identity percentage between genes_id1 and genes_id2.
overlap	overlap	INTEGER	The overlap length between genes_id1 and genes_id2.
s1_start	start_position1	INTEGER	The start position of the alignment in genes_id1.
s1_end	end_position1	INTEGER	The end position of the alignment in genes_id1.
s2_start	start_position2	INTEGER	The start position of the alignment in genes_id2.
s2_end	end_position2	INTEGER	The end position of the alignment in genes_id2.
best1	best_flag_1to2	INTEGER	The flag that indicates the best hit from genes_id1 to genes_id
best2	best_flag_2to1	INTEGER	The flag that indicates the best hit from genes_id2 to genes_id1
def1	definition1	VARCHAR(1000)	The definition for the genes_id1 column.
def2	definition2	VARCHAR(1000)	The definition for the genes_id2 column.

Table 35. SSDB rows from the SSDB database (continued)

version 2.3 column name	version 3.2 column name	Column data type	Description
length1	length1	INTEGER	The length of the amino acid in the genes_id1 column.
length2	length2	INTEGER	The length of the amino acid in the genes_id2 column.

Motif rows

Motif rows are new in the version 3.2 KEGG user-defined functions. The following table describes columns that are returned with motif rows:

Table 36. Motif rows from the SSDB database

Column name	Column data type	Description
motif_id	VARCHAR (100)	The identifier for the motif.
definition	VARCHAR (1000)	The definition for the motif.
genes_id	VARCHAR (100)	The identifier for the gene that contains the motif.
start_position	INTEGER	The start position of the motif_id that matches the query.
end_position	INTEGER	The end position of the motif_id that matches the query.
score	DOUBLE	The Smith-Waterman score of the motif in the TIGRFAM and PROSITE databases that matches the query. The data type of the score in the PROSITE database is FLOAT.
evalue	DOUBLE	The e-value of the motif in the Pfam database that matches the query.

Definition rows

Definition rows are new in the version 3.2 KEGG user-defined functions. The following table lists the columns that are returned with definition rows:

Table 37. Definition rows from the SSDB database

Column name	Column data type	Description
entry_id	VARCHAR(100)	The identifier for the database entry.
definition	VARCHAR(1000)	The definition for the entry.

Related reference

“Changes to KEGG function arguments” on page 69

KEGG function arguments were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9 and later to support the version 3.2 KEGG API.

“Changes to the SSDB function names and accepted arguments” on page 72

KEGG function arguments for the SSDB database were changed in WebSphere Information Integrator, version 8.2 Fix Pack 9 to support the version 3.2 KEGG API.

IMPORT and EXPORT commands for nicknames

Use the IMPORT command to import data into a nickname from a file and the EXPORT command to export data from a nickname to a file.

IMPORT command

The IMPORT command is supported with the DB2 Family, Informix, Microsoft SQL Server, ODBC, Oracle, Sybase, and Teradata data sources.

The IMPORT command for a DB2 for VM remote table does not work correctly for binary data (FOR BIT DATA) columns.

The following restrictions apply when you use the IMPORT command to import data into a nickname:

- The remote object on which the nickname is defined must be a table. You cannot import into a nickname that is defined on a view or synonym.
- The supported file types are IXF, ASC, and DEL.
- The ALLOW WRITE ACCESS clause must be specified. This clause invokes the online import mode. The ALLOW WRITE ACCESS clause allows concurrent applications to have read and write access to the import target table.
- You cannot use the COMMITCOUNT AUTOMATIC mode with nicknames.
- You must specify a value for *n* for the COMMITCOUNT option. The value of *n* cannot be zero.
- Only the INSERT and INSERT_UPDATE operations are supported with nicknames.
- The column types that are not supported with nicknames are Datalink, LOBs, and generated columns. To import LOB data from a remote table, the corresponding nickname column must be a VARCHAR data type.
- The following file-type modifiers are not supported with nicknames:
 - dldelfiletype
 - generatedignore
 - generatedmissing
 - identityignore
 - identitymissing
 - indexixf
 - indexschema
 - lobsinfile
 - nodefaults
 - no_type_idfiletype
 - usedefaults
- Hierarchy (typed table) is not supported with nicknames.

If you submit an IMPORT command that does not adhere to these restrictions, you will receive an SQL error code -27999, for example: SQL27999N The requested IMPORT operation into a remote target (*nickname*) cannot be performed.

The following example uses the INSERT option to import data from an IXF file type:

```
IMPORT FROM import_file_1.ixf OF IXF
ALLOW WRITE ACCESS
COMMITCOUNT 20
INSERT INTO NICKNAME_1;
```

The following example uses the INSERT_UPDATE option to import data from a DEL file type:

```
IMPORT FROM import_file_1.del OF DEL
ALLOW WRITE ACCESS
COMMITCOUNT 50
INSERT_UPDATE INTO NICKNAME_1;
```

The following example uses the INSERT option to import data from an ASC file type. The example includes the STRIPTBLANKS file modifier to truncate any trailing blank spaces in the data. The **METHOD L** parameter specifies the start and end of the column numbers.

```
IMPORT FROM import_file_1.asc OF ASC MODIFIED BY STRIPTBLANKS
METHOD L(1 6, 8 32,34 44, 46 48)
ALLOW WRITE ACCESS
COMMITCOUNT 20
INSERT INTO NICKNAME_1;
```

EXPORT command

You can use the EXPORT command to export data from a nickname to a file. The EXPORT command is supported with the DB2 Family, Informix, Microsoft SQL Server, ODBC, Oracle, Sybase, and Teradata data sources.

The following restrictions apply when you use the EXPORT command to export data from a nickname:

- The description of the target table that is necessary to perform the Import CREATE operation will not be saved in the IXF file format. Use the db2look utility to collect the information that you need to re-create the table.
- The supported file types are IXF and DEL.

Related reference

IMPORT Command

EXPORT Command

Chapter 6. Migrating a replication environment

You can migrate existing version 8 replication environments from DB2 Universal Database on Linux, UNIX, and Windows and WebSphere Information Integrator Replication Edition to version 9 SQL replication.

For information about migrating DB2 and WebSphere Information Integration replication environments, see [Migrating to Replication version 9](#).

Accessing information about IBM

IBM has several methods for you to learn about products and services.

You can find the latest information on the Web at www.ibm.com/software/data/integration/db2ii/support.html :

- Product documentation in PDF and online information centers
- Product downloads and fix packs
- Release notes and other support documentation
- Web resources, such as white papers and IBM Redbooks™
- Newsgroups and user groups
- Book orders

To access product documentation, go to this site:

publib.boulder.ibm.com/infocenter/db2help/topic/

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order.
- To order publications by telephone in the United States, call 1-800-879-2755.

To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide.

Contacting IBM

You can contact IBM by telephone for customer support, software services, and general information.

Customer support

To contact IBM customer service in the United States or Canada, call 1-800-IBM-SERV (1-800-426-7378).

Software services

To learn about available service options, call one of the following numbers:

- In the United States: 1-888-426-4343
- In Canada: 1-800-465-9600

General information

To find general information in the United States, call 1-800-IBM-CALL (1-800-426-2255).

Go to www.ibm.com for a list of numbers outside of the United States.

Accessible documentation

Documentation is provided in XHTML format, which is viewable in most Web browsers.

XHTML allows you to view documentation according to the display preferences that you set in your browser. It also allows you to use screen readers and other assistive technologies.

Syntax diagrams are provided in dotted decimal format. This format is available only if you are accessing the online documentation using a screen reader.

Providing comments on the documentation

Please send any comments that you have about this information or other documentation.

Your feedback helps IBM to provide quality information. You can use any of the following methods to provide comments:

- Send your comments using the online readers' comment form at www.ibm.com/software/awdtools/rcf/.
- Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the information (if applicable). If you are commenting on specific text, please include the location of the text (for example, a title, a table number, or a page number).

Notices and trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM trademarks and certain non-IBM trademarks are marked at their first occurrence in this document.

See <http://www.ibm.com/legal/copytrade.shtml> for information about IBM trademarks.

The following terms are trademarks or registered trademarks of other companies:

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel®, Intel Inside® (logos), MMX and Pentium® are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names might be trademarks or service marks of others.

Index

Numerics

- 32-bit to 64-bit migration
 - UNIX 26

A

- abstract data types
 - DataJoiner restrictions 7
- accessibility 82
- AIX operating system
 - DataJoiner upgrades 8
- applications
 - statement-level isolation 33

B

- BLAST
 - supported versions 2
- BLAST nicknames
 - changes to fixed columns 68

C

- catalog conversion
 - DataJoiner 52
- code page support 8
- commands
 - DB2ILIST 17
 - EXPORT 76
 - IMPORT 76
 - LIST DATABASE DIRECTORY 16
 - LIST DCS DIRECTORY 16
 - LIST NODE DIRECTORY 16
 - REBIND 30
- comments on documentation 82
- configuration settings
 - copying 15
 - restoring after migration 29
- connections
 - restoring for DB2 family data sources 34
- contacting IBM 81
- CTLIB wrapper 9

D

- data sources
 - accessing 40, 41, 44, 46, 47
 - DB2 family 36
 - Informix 38
 - changing wrapper configurations 31
 - configuring after migration 29
 - Microsoft SQL Server 40
 - ODBC 41
 - Oracle 44
 - Sybase 46
 - Teradata 47

- data type mappings
 - DB2 data sources
 - iSeries 55
 - Linux, UNIX, Windows 56
 - VM and VSE 57
 - z/OS and OS/390 54
 - Microsoft SQL Server 62
 - ODBC data sources 64
 - Oracle data sources 58
 - Sybase data sources 66
 - Teradata data sources 67
- database information
 - recording 16
- database objects
 - migrated to federated systems 6
- DataDirect SequeLink ODBC driver 9
- DataJoiner
 - catalog conversion information 52
 - data type mappings
 - Informix data sources 60
 - database path location 9
 - DataJoiner 9
 - migration overview 1
 - migration restrictions 7
 - protocol support
 - DB2 family data sources 53
 - Informix data sources 60
 - Microsoft SQL Server data sources 61
 - ODBC data sources 63
 - Oracle data sources 58
 - Sybase data sources 65
 - Teradata data sources 67
 - terminology 51
- DB2 data sources
 - iSeries
 - data type mappings 55
 - Linux, UNIX, Windows
 - data type mappings 56
 - VM and VSE
 - data type mappings 57
 - z/OS and OS/390
 - data type mappings 54
- DB2 database systems
 - migration overview 1
- DB2 family data sources
 - accessing after migration 36
 - DataJoiner protocol support 53
 - restoring connections after migration 34
- DB2 for iSeries
 - supported versions 2
- DB2 for Linux, UNIX and Windows
 - supported versions 2
- DB2 for VM and VSE
 - supported versions 2
- DB2 for z/OS
 - supported versions 2
- DB2_STATEMENT_ISOLATION server
 - option 33
- db2ilist command 17

- DB2RA protocol 7
- DBLIB wrapper 9
- DDL statements
 - changes from DataJoiner syntax 12
- documentation
 - accessible 82
 - ordering 81
 - Web site 81

E

- Entrez
 - supported versions 2
- environment variable settings
 - copying 15
- Excel files
 - supported versions 2
- EXPORT command 76
- extended indexes
 - restrictions 7

F

- federated stored procedures
 - restrictions 12
- federated systems
 - configuring after migration 29
- federated.sql file 32
- flat files
 - See also table-structured files 2

G

- global catalog
 - DataJoiner catalog conversions 52

H

- HMMER data source
 - supported versions 2

I

- IMPORT command 76
- index conversion
 - type-1 to type-2 30
- indexes
 - converting type-1 indexes to type-2 indexes 30
- Informix
 - supported versions 2
- Informix data sources
 - accessing after migration 38
 - data type mappings 60
 - DataJoiner protocol support 60
- instance information
 - recording 17

K

- KEGG data source
 - supported versions 2
- KEGG user-defined functions
 - changes to arguments 70
 - changes to SSDB columns 73
 - changes to SSDB function names 72

L

- large object data types
 - allowing write operations 11
 - restrictions 11, 14
 - write operations 35
 - write operations on 34
- legal notices 83
- LONG data types
 - restrictions 10, 14

M

- Microsoft Excel
 - See Excel files 2
- Microsoft SQL Server
 - supported versions 2
 - wrapper name changes 8
- Microsoft SQL Server data sources
 - accessing after migration 40
 - data type mappings 62
 - DataJoiner protocol support 61
- migration
 - 32-bit to 64-bit systems 26
 - DataJoiner restrictions 7
 - planning 1
 - replication environments 79
 - spatial data 11
 - WebSphere Information Integrator, version 8.2
 - Linux and UNIX 23
 - Windows 19
- migration paths
 - supported by WebSphere Federation Server 1

N

- nicknames
 - altering column names for XML data sources 33
 - DB2 aliases restrictions 10
 - EXPORT command 76
 - IMPORT command 76
 - trigger restrictions 13
- node information
 - recording 16
- nonvalid data type mappings 10
- nonvalid server options 10

O

- objects
 - migrated to federated systems 6
- ODBC
 - supported versions 2

- ODBC data sources
 - accessing after migration 41
 - data type mappings 64
 - DataJoiner protocol support 63
- OLE DB
 - supported versions 2
- Oracle
 - LONG data types 14
- Oracle data sources
 - accessing after migration 44
 - changing wrapper configuration 31
 - DataJoiner protocol support 58
- NET8
 - data type mappings 58
- Oracle nicknames
 - migrating LONG data types 7

P

- packages
 - rebinding 30
- planning for migration 1
- preparing for migration 15

R

- readers' comment form 82
- REBIND command 30
- replication
 - large object data types 11
 - migration 79
 - restrictions 11
- restrictions 12
 - DataJoiner migration 7
 - Run-time Client 9
 - WebSphere Information Integrator 14
- Run-time Client restrictions 9

S

- screen readers 82
- Script
 - supported versions 2
- SDE transfer files
 - exporting data 11
- server options
 - DB2_STATEMENT_ISOLATION 33
 - restrictions 10
 - two-phase commit 13
- spatial data
 - migration 11
- SQL Replication
 - migration 79
- SQL statements
 - in federated.sql file 32
- SQL syntax
 - DDL changes from DataJoiner 12
- SQLNET wrapper 12
- statement-level isolation
 - behavior in applications 33
- stored procedure nicknames
 - restrictions 12
- Sybase
 - supported versions 2
- Sybase data sources
 - accessing after migration 46

- Sybase data sources (*continued*)
 - changing wrapper configurations 31
 - data type mappings 66
 - DataJoiner protocol support 65
 - wrapper support 9
- Sybase interfaces file
 - copying 15

T

- table-structured files
 - supported versions 2
- Teradata
 - modifying wrapper configurations 49
 - recreating wrapper configurations 50
- Teradata data sources
 - accessing after migration 47
 - data type mappings 67
 - DataJoiner protocol support 67
- terminology
 - DataJoiner and federation server 51
- trademarks 85
- triggers
 - nickname restrictions 13
- two-phase commit
- transactions 13

U

- User-defined type mapping
 - on Oracle LONG columns 13

W

- WebSphere Federation Server
 - migration overview 1
 - migration planning 1
 - overview of migration tasks 1
 - preparing for migration 15
 - supported data sources 1
 - supported migration paths 1
- WebSphere Information Integrator
 - migration overview 1
 - migration restrictions 14
 - restoring configuration settings 29
- WebSphere Information Integrator, version 8.2
 - migration
 - Linux and UNIX 23
 - Windows 19
- WITH CHECK OPTION clause 12
- wrapper names
 - DB2 family data sources 53
 - Informix data sources 60
 - Microsoft SQL Server data sources 61
 - ODBC data sources 63
 - Oracle data sources 58
 - Sybase data sources 65
- wrappers
 - changing configurations for data sources 31
- write operations
 - large objects 35

X

XML

supported versions 2

XML nicknames

altering nickname column names 33



Printed in USA

SC19-1019-00

