SAP BusinessObjects

Upgrade Guide

- SAP BusinessObjects Data Services 4.1 Support Package 1 (14.1.1.0)

2012-12-12

2012-12-12

Contents

Welcome to SAP BusinessObjects Data Services

1.1 Welcome

SAP BusinessObjects Data Services delivers a single enterprise-class solution for data integration, data quality, data profiling, and text data processing that allows you to integrate, transform, improve, and deliver trusted data to critical business processes. It provides one development UI, metadata repository, data connectivity layer, run-time environment, and management console—enabling IT organizations to lower total cost of ownership and accelerate time to value. With SAP BusinessObjects Data Services, IT organizations can maximize operational efficiency with a single solution to improve data quality and gain access to heterogeneous sources and applications.

1.2 Documentation set for SAP BusinessObjects Data Services

You should become familiar with all the pieces of documentation that relate to your SAP BusinessObjects Data Services product.

Document	What this document provides
Administrator's Guide	Information about administrative tasks such as monitoring, lifecycle management, security, and so on.
Customer Issues Fixed	Information about customer issues fixed in this release.
Designer Guide	Information about how to use SAP BusinessObjects Data Services Designer.
Documentation Map	Information about available SAP BusinessObjects Data Services books, languages, and locations.
Installation Guide for Windows	Information about and procedures for installing SAP BusinessObjects Data Services in a Windows environment.
Installation Guide for UNIX	Information about and procedures for installing SAP BusinessObjects Data Services in a UNIX environment.
Integrator's Guide	Information for third-party developers to access SAP BusinessObjects Data Services functionality using web services and APIs.

Document	What this document provides
Master Guide	Information about the application, its components and scenarios for planning and designing your system landscape. Information about SAP BusinessObjects Information Steward is also provided in this guide.
Management Console Guide	Information about how to use SAP BusinessObjects Data Services Administrator and SAP BusinessObjects Data Services Metadata Reports.
Performance Optimization Guide	Information about how to improve the performance of SAP BusinessObjects Data Services.
Reference Guide	Detailed reference material for SAP BusinessObjects Data Services Designer.
Release Notes	Important information you need before installing and deploying this version of SAP BusinessObjects Data Services.
Technical Manuals	A compiled “master” PDF of core SAP BusinessObjects Data Services books containing a searchable master table of contents and index: • *Administrator's Guide* • *Designer Guide* • *Reference Guide* • *Management Console Guide* • *Performance Optimization Guide* • *Supplement for J.D. Edwards* • *Supplement for Oracle Applications* • *Supplement for PeopleSoft* • *Supplement for Salesforce.com* • *Supplement for Siebel* • *Supplement for SAP* • *Workbench Guide*
Text Data Processing Extraction Customization Guide	Information about building dictionaries and extraction rules to create your own extraction patterns to use with Text Data Processing transforms.
Text Data Processing Language Reference Guide	Information about the linguistic analysis and extraction processing features that the Text Data Processing component provides, as well as a reference section for each language supported.
Tutorial	A step-by-step introduction to using SAP BusinessObjects Data Services.
Upgrade Guide	Release-specific product behavior changes from earlier versions of SAP BusinessObjects Data Services to the latest release. This manual also contains information about how to migrate from SAP BusinessObjects Data Quality Management to SAP BusinessObjects Data Services.
What's New	Highlights of new key features in this SAP BusinessObjects Data Services release. This document is not updated for support package or patch releases.

Document	What this document provides
Workbench Guide	Provides users with information about how to use the Workbench to migrate data and database schema information between different database systems.

In addition, you may need to refer to several Supplemental Guides.

Document	What this document provides
Supplement for J.D. Edwards	Information about interfaces between SAP BusinessObjects Data Services and J.D. Edwards World and J.D. Edwards OneWorld.
Supplement for Oracle Applications	Information about the interface between SAP BusinessObjects Data Services and Oracle Applications.
Supplement for PeopleSoft	Information about interfaces between SAP BusinessObjects Data Services and PeopleSoft.
Supplement for Salesforce.com	Information about how to install, configure, and use the SAP BusinessObjects Data Services Salesforce.com Adapter Interface.
Supplement for SAP	Information about interfaces between SAP BusinessObjects Data Services, SAP Applications, SAP Master Data Services, and SAP NetWeaver BW.
Supplement for Siebel	Information about the interface between SAP BusinessObjects Data Services and Siebel.

We also include these manuals for information about SAP BusinessObjects Information platform services.

Document	What this document provides
Information Platform Services Administrator's Guide	Information for administrators who are responsible for configuring, managing, and maintaining an Information platform services installation.
Information Platform Services Installation Guide for UNIX	Installation procedures for SAP BusinessObjects Information platform services on a UNIX environment.
Information Platform Services Installation Guide for Windows	Installation procedures for SAP BusinessObjects Information platform services on a Windows environment.

1.3 Accessing documentation

You can access the complete documentation set for SAP BusinessObjects Data Services in several places.

1.3.1 Accessing documentation on Windows

After you install SAP BusinessObjects Data Services, you can access the documentation from the Start menu.

1. Choose **Start > Programs > SAP BusinessObjects Data Services 4.1 > Data Services Documentation > All Guides**.
2. Click the appropriate shortcut for the document that you want to view.

1.3.2 Accessing documentation on UNIX

After you install SAP BusinessObjects Data Services, you can access the documentation by going to the directory where the printable PDF files were installed.

1. Go to `<LINK_DIR>/doc/book/en/`.
2. Using Adobe Reader, open the PDF file of the document that you want to view.

1.3.3 Accessing documentation from the Web

You can access the complete documentation set for SAP BusinessObjects Data Services from the SAP BusinessObjects Business Users Support site.

To do this, go to http://help.sap.com/bods.

You can view the PDFs online or save them to your computer.

1.4 SAP BusinessObjects information resources

A global network of SAP BusinessObjects technology experts provides customer support, education, and consulting to ensure maximum information management benefit to your business.

Useful addresses at a glance:

Address	Content
Customer Support, Consulting, and Education services http://service.sap.com/	Information about SAP Business User Support programs, as well as links to technical articles, downloads, and online forums. Consulting services can provide you with information about how SAP BusinessObjects can help maximize your information management investment. Education services can provide information about training options and modules. From traditional classroom learning to targeted e-learning seminars, SAP BusinessObjects can offer a training package to suit your learning needs and preferred learning style.
Product documentation http://help.sap.com/bods/	SAP BusinessObjects product documentation.
Supported Platforms (Product Availability Matrix) https://service.sap.com/PAM	Get information about supported platforms for SAP BusinessObjects Data Services. Use the search function to search for Data Services. Click the link for the version of Data Services you are searching for.

About this guide

Use this guide to help you with upgrading to Data Services 4.1.

This document contains the following upgrade topics:

- Upgrading from previous versions of Data Services.
- Behavior changes associated with each version of the Data Integrator and Data Services products.
- Migration of your Data Quality XI projects into Data Services.

Changes in Data Services 4.1 SP1

The Data Services 4.1 SP1 release, like the Data Services 4.0 and 4.1 releases, continues to use central user management, which relies on SAP BusinessObjects Business Intelligence platform. Users are authenticated against the BI platform security, and repositories are registered in SAP BusinessObjects BI platform.

You can use either an existing SAP BusinessObjects Business Intelligence platform 4.0 SP4 installation to manage Data Services users and repositories, or SAP BusinessObjects Information platform services 4.0 SP4, which is availabe to all Data Services customers and contains the required BI platform services for user management.

For information about installing SAP BusinessObjects Information platform services, see the *SAP BusinessObjects Information platform services Guide.*

In SAP BusinessObjects Data Services 4.0 and 4.1, significant changes were made in the following areas. These changes will affect those who are upgrading from Data Services 3.x (or Data Integrator 11.7.x).

- **System requirements**. For the latest information about system requirements, see the *Product Availability Matrix* document located in the SAP Service Marketplace: https://service.sap.com/PAM.
- **Installation**. For details, see the *Installation Guide*.
- **Universal Data Cleanse**. If you currently use Universal Data Cleanse and have modified a person and firm dictionary or created a custom dictionary, ensure that the Cleansing Package Builder in Information Steward is available before you upgrade. You will need Cleansing Package Builder to migrate your dictionary file to the new cleansing package format. For more information, see Data Cleanse solution.
- **Security**. By default, you are now required to enter a username and password for your Data Services repository when you access the Designer or open a Data Quality report. See the Administrator's Guide for more information.
- **SAP customers**. This version includes several enhancements to functions, procedures, and authorizations that provide secure integration with SAP systems. You can now at a more granular level assign authorizations that can limit which users can execute functions and which programs can be started. In addition, the new secured functions are now delivered into a new /BODS/ namespace instead of the customer Z namespace. For details, see the *Supplement for SAP*.

  Note:
  You must update the Data Services-supplied transport files and update the SAP authorizations to the 4.1 version. For details, see the *Supplement for SAP*.

Installing Data Services 4.1 SP1

Refer to the *SAP BusinessObjects Data Services Installation Guide* for detailed, platform-specific installation instructions.

The installer migrates the objects in the `dsconfig.txt` file to the new SAP BusinessObjects Data Services 4.1 SP1 locations. For more information about these changes, see User Access Control (UAC) support.

Installing to a different machine

If you are installing Data Services 4.1 SP1 to a different machine than the previous version, copy the `dsconfig.txt` file to the new server manually before you run the install. When prompted by the installation program for a previous configuration, provide the location of the `dsconfig.txt` file.

The Upgrade Manager does not migrate the job server and access server configuration. If you install Data Services 4.1 SP1 to a different server than Data Services XI 3.x, Data Services or Data Integrator 11.7, use the Server Manager after installation to configure the Job Server. If Access Servers were configured for the previous version, you must use the Server Manager to configure them again.

Upgrading to Data Services 4.1 SP1

There are three paths to upgrading to SAP BusinessObjects Data Services 4.1:

- Upgrading from Data Services XI 3.2 and prior versions, or Data Integrator 11.7.x
- Upgrading from Data Services 4.0
- Upgrading from Data Services 4.1

The following sections help you upgrade from either of these paths.

Upgrade considerations

Whether you are upgrading from Data Services 4.0 or some other prior version, keep in mind the following, because they apply to both upgrade scenarios.

- Data Services Profiler configuration: When you upgrade to Data Services 4.1 SP1 (14.1.1), the Profiler configuration parameters are set to default values. If you had modified any of the Profiler configuration parameters, after installing 4.1 SP1, you need to log on to the Management Console and change the Profiler configuration parameters to your custom settings.
- Oracle users: Before upgrading to version 14.1.1 be sure to grant the database account the create any sequence privilege.
- SAP Applications users: You must update the Data Services-supplied transport files and update the SAP authorizations to the 4.1 version. For details, see the *Supplement for SAP*.

5.1 Upgrading from Data Services 4.1

You can install Data Services 4.1 SP1 on top of an existing Data Services 4.1 installation. You do not need to uninstall Data Services 4.1 first.

To use Data Services 4.1 SP1, you will need to upgrade your repository. For more information, see the Data Services*Administrator Guide*.

5.2 Upgrading from Data Services 4.0

You can install Data Services 4.1 on top of an existing Data Services 4.0 installation. You do not need to uninstall Data Services 4.0.

You must upgrade Business Intelligence platform or Information platform services to the compatible version. Refer to SAP Note 1723342 for possible upgrade scenarios.

Be aware of the following when installing Data Services 4.1 over Data Services 4.0:

- You will not be able to change the features that you installed with Data Services 4.0.
- Your 4.0 CMS configurations will not change.
- You have the ability to reuse configurations (`DSConfig.txt` files).
- You must upgrade the 4.0 local and central repositories using the Repository Manager after installing Data Services 4.1.

  For more information, see the *SAP BusinessObjects Data Services Administrators Guide*.

- If you configured a MySQL or Oracle repository at any time after installing Data Services 4.0, you must manually configure the JDBC drivers after upgrading to Data Services 4.1.

  For more information, see “Post-installation, Configuring JDBC drivers for SAP HANA, Oracle, and MySQL” in the *SAP BusinessObjects Data Services Installation Guide*.

5.3 Upgrading from Data Services 3.2 and prior

The SAP BusinessObjects Data Services 4.1 upgrade process consists of the following steps:

1. Before proceeding with the upgrade process, back up configuration files and repositories from previous releases (Data Services XI 3.x or Data Integrator 11.7), as well as Data Cleanse files from Data Services XI 3.2.
2. Uninstall the previous version of Data Services XI 3.x or Data Integrator 11.7.
3. Make sure that you either have installed or upgraded to SAP BusinessObject Business Intelligence platform 4.0 FP3 or have installed SAP BusinessObjects Information platform services 4.0 FP3.
4. Install the SAP BusinessObjects Data Services 4.1 product. For more information, see the *SAP BusinessObjects Data Services Installation Guide*.
5. Upgrade the local and central repositories using the repository manager. For more information, see the *SAP BusinessObjects Data Services Administrators Guide*.
6. If you are upgrading from Data Services XI 3.x or Data Integrator 11.7, migrate the users, repositories, and other objects from `admin.xml` and `sapconnections.xml` to the SAP BusinessObjects Business Intelligence platform by running the Upgrade Manager.
7. Migrate the real-time services configuration, contained in the `as.xml` file to the new Data Services 4.1 location.
8. Restart the web application server.

These steps are explained in detail in the following sections.

Restriction:
You can import your Data Services XI 3.2 Person_Firm_* cleansing package into the new single global 4.1 cleansing package version using Cleansing Package Builder 4.1. You can also merge your 4.0 Person_Firm_* cleansing package changes into the single global 4.1 cleansing package. To import or

merge your changes, you will need to install SAP BusinessObjects Information Steward to get the Cleansing Package Builder functionality.

5.3.1 Backing up repositories, configuration files, and Data Cleanse files

Before you install SAP BusinessObjects Data Services 4.1, back up your Data Services XI 3.x or Data Integrator 11.7 repositories, configuration files, and Data Cleanse 3.2 files.

1. Back up your local and central repositories. Create a database backup of the schema or database where the Data Services 3.x or Data Integrator 11.7 repository is located. The exact backup steps differ between database vendors. Make sure that your database backup allows you to restore the all of the repository tables and data.
2. Back up the following configuration files using the RDBMS database utilities:
   - `admin.xml`
   - `sapconnections.xml`
   - `as.xml`
   - `dsconfig.txt`
3. Back up your Data Cleanse 3.2 files.

MySQL

The MySQL database that was bundled with Data Services 3.x and older is uninstalled when Data Services is uninstalled. Special care is needed when you upgrade the repository to Data Services 4.1. You can use either of the following options to upgrade the repository.

Option 1

1. Before you uninstall Data Services XI 3.x or Data Integrator 11.7, back up the current MySQL data folder.
2. Obtain a new copy of MySQL 5.0 (not 5.1) and install it.
3. Copy the backup MySQL data folder to the new MySQL 5.0 location.
4. Uninstall Data Services XI 3.x or Data Integrator 11.7. This also uninstalls the previous version of the bundled MySQL.
5. Install Data Services 4.1.
6. Upgrade the new MySQL 5.0 repository to version 14.1.0.

Option 2

1. Before you uninstall Data Services XI 3.x or Data Integrator 11.7, install Data Services 4.1 on a new machine.
2. Install MySQL ODBC Connector (3.51.23, 3.51.24, or 3.51.25) on the new machine where Data Services 4.1 is installed.
3. Run the Data Services 4.1 Repository Manager to upgrade the previous MySQL repository to version 14.1.0.0.
4. Export the MySQL repository to an ATL file (version 14.1.0.0).
5. Import the ATL file into a repository in any supported database (either the bundled SQL Server Express in Information platform services 4.0 FP3 or any other database).

6. Uninstall Data Services XI 3.x or Data Integrator 11.7. This also uninstalls the previous version of the bundled MySQL.

5.3.2 Uninstalling the previous version of Data Services or Data Integrator

Before you install SAP BusinessObjects Data Services 4.1, you must uninstall your current version of Data Services XI 3.x or Data Integrator 11.7.x .

For details, see the *Installation Guide* for the version that you are using. For any known issues and limitations regarding uninstalling, see the appropriate *Release Notes* for the version of your deployment.

Related Topics

- Installing Data Services 4.1 SP1

5.3.3 Upgrading repositories

You must upgrade local and central Data Services repositories to register them in the SAP BusinessObjects Business Intelligence platform or Information platform servicesCentral Management Console (CMC). You must individually upgrade each of the repositories that are listed in the Data ServicesManagement Console Administrator.

Use the Data ServicesRepository Manager to upgrade the repositories.

When you upgrade a secure central repository, you must assign a local repository that can be used to check out, check in, or get latest objects from the secure central repository.

For more information, see the *SAP BusinessObjects Data Services Administrator's Guide*.

5.3.4 Migrating users, repositories, and other objects

Data Services now uses the SAP BusinessObjects Business Intelligence platform for user and repository configuration information. If you are upgrading from Data Services XI 3.x or Data Integrator 11.7, you must migrate this information and other objects to SAP BusinessObjects Business Intelligence platformCentral Management Server (CMS) repository objects.

For previous versions of Data Services, the `admin.xml` and `sapconnections.xml` files were used by the Data ServicesManagement Console Administrator. They were located in the `<LINK_DIR>/conf` folder.

The admin.xml file contained the following configuration information, if applicable:

- Repositories (local and central)
- Users
- Central repository users
- Access Server configuration
- CMS connection information (where Data Services job schedules are stored)
- Web service enabled jobs configuration

The sapconnections.xml file contained the following configuration information:

- RFC connections

Beginning with Data Services 4.0, use of admin.xml and sapconnections.xml files is discontinued for storing users, repositories, access server information, and RFC configurations. However, the admin.xml file continues to exist and is used to store other configurations for the Data ServicesManagement Console Administrator.

Migrating ABAP data flows

To use the new RFC Data transfer method for existing ABAP data flows in datastores that use the ABAP execution option **Execute_preloaded**, you must regenerate the ABAP code for each data flow and upload it to the SAP server. To generate ABAP code, open the ABAP data flow and select **Validation > Generate ABAP code**. For more information, see the *Supplement for SAP*.

5.3.4.1 Using Upgrade Manager

A command-line Upgrade Manager tool is provided to migrate users, repositories, and other objects from the admin.xml and sapconnections.xml files.

On Windows, use the DSXI40UpgradeMgr.bat file. On UNIX, use the DSXI40UpgradeMgr.sh file. These files are installed in the <*LINK_DIR*>/bin folder.

Prerequisites

The Upgrade Manager checks the following conditions before running:

- SAP BusinessObjects Data Services 4.1 is installed correctly and integrated with SAP BusinessObjects Business Intelligence platform or Information platform services.
- New SAP BusinessObjects Data Services 4.1 LINK_DIR environment variable is set to the correct Data Services 4.1 install location.
- An SAP BusinessObjects Business Intelligence platform CMS user ID and password with administrator privileges is used.
- The admin.xml and sapconnections.xml files are accessible in the location where Upgrade Manager is run.

Migrating all or separate objects

The Upgrade Manager can migrate objects from the `admin.xml` and `sapconnections.xml` files all at once or one type at a time (for example, repositories, users, and so on), as explained in the following sections.

Log file

Upgrade Manager creates a detailed run-time log file (`UpgradeMgr.log`) that lists error messages. It is available in the `<LINK_DIR>/log` folder. The error messages are also written to the Management Console. If any of the required conditions are not satisfied, Upgrade Manager does not run the migration process. The log file provides details if the condition check fails.

You can run Upgrade Manager multiple times to complete the migration. You should correct any errors in the tool log before running Upgrade Manager again.

Restart the web application server

After Upgrade Manager successfully completes processing, you must restart the web application server.

Related Topics

- Migrating all objects
- Migrating separate objects

5.3.4.2 Migrating all objects

Upgrade Manager can migrate all of the configuration objects from the `admin.xml` and `sapconnections.xml` files at once.

Example: To migrate all objects

1. To migrate all objects from `admin.xml` and `sapconnections.xml`, including repositories, users, and so on, use the following command as an example:

   ```
   DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf"
   -m -a
   ```

   The -m option is the command to migrate. The -a option migrates all of the configuration objects to the new SAP BusinessObjects Data Services 4.1 environment.
2. Check the log file for errors and correct them before running Upgrade Manager again. For example, if, after you run Upgrade Manager, only the repositories remain to be migrated, then you should run the tool with the appropriate options (in this example, -m -r), so that it does not attempt to migrate other objects. Upgrade Manager reports an error if the configuration object already exists in SAP BusinessObjects Business Intelligence platform, skips that object, and continues to the next one.
3. After Upgrade Manager successfully completes processing, restart the web application server.

5.3.4.3 Migrating separate objects

To migrate one type of object at a time (for example, repositories, users, and so on), see the following sections.

5.3.4.3.1 Command-line options

This section describes the command-line options that the Upgrade Manager supports for different migration scenarios. The command-line options are case sensitive. The Upgrade Manager depends on the environment variable `LINK_DIR` to get the dependent `.jar` files.

Processing options

The following table describes processing options. Only one processing option can be used at a time; otherwise, Upgrade Manager reports an error.

Option	Description	Required
`-m`	Performs the migration.	No
`-l`	Lists all of the objects to be migrated from the `admin.xml` and `sapconnections.xml` files. Results are output to the Management Console.	No
`-v`	Compares the contents in the `admin.xml` and `sapconnections.xml` files with what already exists in the CMS. Results are output to the Management Console.	No
`-h`	Help (displays usage).	No

Object migration options

The following table describes options that migrate all or separate objects. The `-a` option migrates all objects. The `-c`, `-p`, `-r`, `-s`, `-u`, `-x`, and `-w` options let you migrate one type of object at a time. All of these options must be used with the `-m` option.

Option	Description	Required
`-a`	Migrates all objects.	No

Option	Description	Required
-c	Migrates all or specific central repository users. Separate multiple users with a comma. To migrate all central repository users: -c To migrate specific central repository users: -cuser1,-user2	No
-p	Migrates password file and schedules.	No
-r	Migrates all or specific repositories. Separate multiple repositories with a comma. To migrate all repositories: -r To migrate specific repositories: -rrepo1,repo2	No
-s	Migrates RFC information.	No
-u	Migrates all or specific users. Separate multiple users with a comma. To migrate all users: -u To migrate specific users: -uuser1,user2	No
-x	Migrates access servers.	No
-w	Migrates web service job configurations.	No

Connection options

The following table options that contain connection information.

Option	Description	Required
-CC	The Central Management System (CMS) server and port number where the Data Services application is registered, separated by a colon: *servername:portnumber*. The default port number is 6400. -CCserver1:6400	Yes
-CU	CMS user name. -CUcmsuser	Yes

Option	Description	Required
-CP	CMS password. -CPcmspass	Yes
-CA	CMS authentication mode. Possible values are: • secEnterprise (default) • secLDAP • secWINAD -CAsecEnterprise	No
-LO	Location of the previous version of Data Services or Data Integratoradmin.xml and sapconnections.xml files. The files can be in any location on the new server, as long as the server can access them. -LO"C:\Program Files\Business Objects\Data Services\conf"	Yes
-FH	Profiler server host. If you have migrated profiler repositories, use this option to indicate the new profiler server host. If not provided, localhost is used as the default. -FHprofserver	No
-FP	Profiler server port. Defaults to 8080. Used for profiler repository. -FP9999	No

5.3.4.3.2 Command-line arguments in a text file

The command-line arguments for the command script can also be passed from a text file. The arguments can be listed in a text file with one argument per line.

Example: To pass command-line arguments in a text file

A text file, args.txt, contains the following arguments to migrate repositories:

```
#parameters in a text file
-CCserver1
-CUboeuser
-CPpassword
```

```
-LO"C:\xi3conf"

-m

-r
```

Calling Upgrade Manager with the argument file `DSXI40UpgradeMgr.bat args.txt` from a command prompt has the same effect as calling arguments in a command line. Lines beginning with the # symbol contain comments.

5.3.4.3.3 Migrating repositories

Upgrade Manager does not migrate the SAP BusinessObjects Data Services repositories; rather, it migrates the repository information from the `admin.xml` file to the SAP BusinessObjects EnterpriseCentral Management Server (CMS). Before you run Upgrade Manager, upgrade all repositories listed in the `admin.xml` file or the Data Services Management Console.

Note:
Data Services 3.2 non-secure central repositories must be manually registered in the Central Management Console.

When you run Upgrade Manager, it checks the repository version before running the migration process. If the repository has not been upgraded with the Repository Manager, repository migration is skipped during the migration process with appropriate messages. If the repositories were skipped during the migration process, you can run Upgrade Manager again with the repository migration option `-r`.

By default, the repository owner is the user that runs Upgrade Manager.

Example: To migrate all repositories

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-r
```

Example: To migrate a specific repository

In this example, the repository name is dsrepo.

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-rdsrepo
```

The `-r` option allows comma-separated values; for example, `-rdsrepo1,dsrepo2`.

Note:
For information about upgrading a repository that was created using the MySQL database that was bundled with previous versions of Data Services, see Backing up repositories, configuration files, and Data Cleanse files .

5.3.4.3.4 Migrating users

Upgrade Manager migrates the users created in the Data ServicesManagement Console (and also displayed in the `admin.xml` file) to SAP BusinessObjects Business Intelligence platform as SAP BusinessObjects Business Intelligence platform users. The user role remains as it was defined in the `admin.xml` file. For example, user *`opr1`* with an *`Operator`* role is migrated to SAP BusinessObjects Business Intelligence platform as user *`opr1`* in the *`Data Services Operator`* group.

Note:
Because SAP BusinessObjects Data Services 4.x significantly changes the user management model, it is recommended that you re-evaluate your user management strategy and consider manually designing a new security model rather than simply migrating your users with Upgrade Manager. This would allow you to take advantage of the new user management features and benefits.

Example: To migrate all users

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-u
```

Example: To migrate a specific user

In this example, the repository name is `opr1`.

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-uopr1
```

The `-u` option allows comma-separated values; for example, `-uopr1,opr2`.

5.3.4.3.5 Migrating central repository users

Upgrade Manager migrates users from the central Data Services repository to SAP BusinessObjects Business Intelligence platform as SAP BusinessObjects Business Intelligence platform users. Upgrade Manager identifies the central repository from the `admin.xml` file, connects to the repository, and migrates the users.

Example: To migrate all users from a central repository

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-c
```

Example: To migrate specific users

In this example, the central repository user names are `cuser1` and `cuser2` .

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-ccusr1,cuser2
```

The `-c` option allows comma-separated values; for example, `-ccusr1,cuser2`.

5.3.4.3.6 Migrating access server configurations

Upgrade Manager migrates access server host and port configurations from the `admin.xml` file to the SAP BusinessObjects Enterprise Central Management Server (CMS) repository.

Example: To migrate all access server configurations

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-x
```

The `-x` option does not take any values.

5.3.4.3.7 Migrating web service job configurations

Upgrade Manager migrates configurations for jobs published as a web service to the Management Console in the Data Services XI 3.x or Data Integrator 11.7 `admin.xml` file to the new Data Services 4.1`admin.xml` file.

Example: To migrate all web service job configurations

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-w
```

The `-w` option does not take any values.

5.3.4.3.8 Migrating RFC connections

Upgrade Manager migrates RFC connection configurations stored in the `sapconnections.xml` file to the SAP BusinessObjects Enterprise Central Management Server (CMS) repository.

Example: To migrate all RFC server configurations

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-s
```

The `-s` option does not take any values.

5.3.4.3.9 Migrating password files and schedules

Upgrade Manager migrates password file and schedules together. The password file must be regenerated because in Data Services XI 3.x and previous releases, the generated password file included the repository connection information. In SAP BusinessObjects Data Services 4.1, the repository connection information is replaced with CMS connection information and the repository name.

How it works

Upgrade Manager connects to the job server to regenerate the password file and/or job command file. Upgrade Manager reads the job server information from the schedule table in each Data Services repository found in the `admin.xml` file.

If SAP BusinessObjects Business Intelligence platform CMS schedules are used, they must be migrated to the new SAP BusinessObjects Business Intelligence platform CMS through the SAP BusinessObjects Business Intelligence platform CMS repository migration process.

Restrictions

If the Data Services 4.1 job server is installed on a different host than Data Services XI 3.x or Data Integrator 11.7, the password file and command file cannot be regenerated with the old server name. In this case, the password files must be generated using the Data ServicesManagement Console repository administration. The schedules (such as Windows AT, UNIX cron, or any third-party scheduling tool) must be manually moved or recreated in the new host.

Example: To migrate and regenerate password files and schedules

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -m
-p
```

The `-p` option does not take any values.

5.3.4.4 Additional Upgrade Manager examples

The following are additional ways that you can use Upgrade Manager.

Example: To list the contents of the `admin.xml` and `sapconnections.xml` files

The following example lists all of the objects to be migrated from the `admin.xml` and `sapconnections.xml` files. The results are output to the Management Console.

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -l
```

Example: To preview the migration

The following example compares the contents in the `admin.xml` and `sapconnections.xml` files with what already exists in the CMS. The results are output to the Management Console.

```
DSXI40UpgradeMgr.bat -CCserver1 -CUboeuser -CPpassword -LO"C:\xi3conf" -v
```

5.3.4.5 Error messages

Any failure during migration is logged in the trace log for analysis. Upgrade Manager also traces all of the critical points in migration. Upgrade Manager returns the following error codes:

Error code	Description	How to correct
0	No error.	
10	Command-line parsing error.	Check the command-line options and values.
11	Data Services 4.1 `LINK_DIR` environment variable is not set.	Verify that the `LINK_DIR` environment value is set correctly for the Data Services 4.1 installation.
12	Directory location of Data Services XI 3.x or Data Integrator 11.7 `admin.xml` and `sapconnections.xml` files are not correct.	Check the `-LO` option value.
13	XML parsing error.	Parsing of `admin.xml` and `sapconnections.xml` files failed. Edit the XML to correct the XML structure problem.
14	CMS login error.	Check the SAP BusinessObjects Business Intelligence platform CMS credentials.
15	Data Services not installed.	Install Data Services application objects in SAP BusinessObjects Business Intelligence platform.
16	File read error.	Error reading the `admin.xml` or `sapconnections.xml` file. Verify that the files have been copied to the correct location (by default, `C:\Program Files\Business Objects\Data Services\conf`), and have read permissions.

Error code	Description	How to correct
100	Data Services 4.1 LINK_DIR environment variable is not set.	Verify that the LINK_DIR environment value is set correctly for the Data Services 4.1 installation.
101	Unable to get the SAP BusinessObjects Business Intelligence platform SDK install directory.	The SAP BusinessObjects Business Intelligence platform SDK .jar files not found. Check the LINK_DIR value.
102	JAVA_HOME not found.	Verify that the LINK_DIR environment value is set correctly.

5.3.4.6 Running Upgrade Manager with SSL-enabled CMS

When the SAP BusinessObjects Business Intelligence platform CMS is SSL-enabled, the following SSL-related parameters are required to log in to the CMS. Provide the parameters when you call Upgrade Manager. Contact your CMS administrator to retrieve the necessary certificates and keys, and copy them to the local machine where SAP BusinessObjects Data Services 4.1 is installed.

- -Dbusinessobjects.orb.oci.protocol=ssl
- -DcertDir=<*SSL certificates directory*>
- -DtrustedCert=<*trusted certificate store*>
- -DsslCert=<*SSL certificate*>
- -DsslKey=<*SSL key*>
- -Dpassphrase=<*passphrase file*>

Example: Windows

In Windows, the double quotes around the JVM arguments -D is necessary due to Windows limitation in the command-line arguments processing.

```
DSXI40UpgradeMgr.bat -D"businessobjects.orb.oci.protocol=ssl" -D"cert
Dir=C:\temp\SSLCert" -D"trustedCert=cacert.der" -D"sslCert=clientcert.der"
-D"sslKey=client.key" -D"passphrase=passphrase.txt" -CCserver1 -CUboeuser
-CPpassword -LO"C:\xi3conf" -v
```

Example: UNIX

```
DSXI40UpgradeMgr.sh -Dbusinessobjects.orb.oci.protocol=ssl -DcertDir=/u/my
home/sslcert -DtrustedCert=cacert.der -DsslCert=clientcert.der -Dss
lKey=client.key -Dpassphrase=passphrase.txt -CCserver1 -CUboeuser -CPpassword
-LO/u/myhome/xi3conf -v
```

5.3.5 Migrating real-time services configurations

After you have re-configured your access server instances using Server Manager, you must migrate your real-time configurations. The real-time services configurations are stored in the `as.xml` file, which is located in the access server folder of the *<LINK_DIR>* directory. When you migrate Data Services XI 3.x, 4.0, or Data Integrator 11.7 real-time services configurations, copy the Data Services 3.x, 4.0, or Data Integrator 11.7 `as.xml` file to the access server folder of the new Data Services 4.1 *<DS_COMMON_DIR>* folder. When the access server starts, it updates the necessary values in this file with the new encryption algorithm.

5.3.6 Verifying the upgrade process

After you successfully run Upgrade Manager and before you verify the migration, restart the application server for the Data ServicesManagement Console.

The following is a list of verification steps for a successful upgrade and migration process.

1. In the SAP BusinessObjects Business Intelligence platformCentral Management Console (CMC) Web application, select **Data Services > Repositories**.

   All of the migrated repositories should be listed.

2. In the SAP BusinessObjects Data ServicesManagement Console, verify the following:
   a. All successfully migrated repositories, jobs and their schedules, access server configurations, and web services configurations are listed in the Administrator page.
   b. The central repository is listed, if applicable.
   c. RFC server configurations are available under SAP connections.

3. Start the Data ServicesDesigner using a migrated user.

   Now that you have completed the migration process, you can select one of the migrated repositories, open the projects, and execute a job.

Data Services Behavior Changes

This chapter describes behavior changes associated with the Data Integrator product since version 11.7.0 and in Data Services since 12.0.0 including Data Quality functionality. Each behavior change is listed under the version number in which the behavior originated. If you are upgrading from a version earlier than Data Integrator 11.7.0, it is recommended that you first upgrade to 11.7.0, and then upgrade to the latest version.

For information about how to migrate your Data Quality projects into Data Services, see Data Quality to Data Services Upgrade.

6.1 Behavior changes in version 14.1.1

The following sections describe changes in the behavior of SAP BusinessObjects Data Services 4.1 Support Package 1 (14.1.1) from previous releases of Data Services and Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see Data Quality to Data Services Upgrade.

6.1.1 Global Address Cleanse transform changes

This section describes the behavior changes in this release for the Global Address Cleanse transform.

Locality1_Name output field changed

For German addresses, the Locality1_Name field has changed to include only the locality information and not the locality addition. A new output field, Locality1_Addition, has been added that contains the locality addition information. For example, for Freiberg am Neckar, Freiberg is output in the Locality1_Name field and am Neckar is output in the Locality1_Addition field. The Locality1_Full output field may or may not include locality addition information depending on the setting of the new Include Locality Addition option.

6.1.2 Universal Data Cleanse transform changes

This section describes the behavior changes in this release for the Universal Data Cleanse transform.

Parsing order changed

The discrete phone, date, and email input fields are parsed before multiline input fields due to the implementation of the "One-to-one mapping" feature. In previous releases, the multiline input fields were parsed first. Due to this change, you might notice a difference in your output when you have a combination of multiline and discrete input fields, even when the "One-to-one mapping" option is set to **No**.

6.1.3 Connection Manager

In this version of Data Services, the Connection Manager replaces the dsdb_setup.sh utility. The Connection Manager now provides the following functionality:

- A **Data Sources** tab to configure the environment variables required for ODBC database types for DSN connections, which replaces the functionality of dsdb_setup.sh.
- A **Drivers** tab to configure the ODBC drivers which will be used in DSN-less connections.

Related Topics

- Administrator Guide: Using the Connection Manager for UNIX systems
- Administrator Guide: DSN-less and TNS-less connections

6.2 Behavior changes in version 14.1.0

The following sections describe changes in the behavior of SAP BusinessObjects Data Services 4.1 (14.1.0) from previous releases of Data Services and Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see Data Quality to Data Services Upgrade.

6.2.1 Metadata Integrator

If you are migrating from SAP BusinessObjects Data Services XI 3.x or 4.0, and have SAP BusinessObjects Business Intelligence platform Metadata Integrator configured to collect from a CMS repository version XI 3.x or 4.0, you must retain the Business Intelligence platform 3.x or 4.0 client machine with Data Services XI 3.x or 4.0 Metadata Integrator installation to continue to collect CMS metadata from Business Intelligence platform XI 3.x or 4.0 systems.

From Data Services 4.1, you cannot directly configure Metadata Integrator to collect from Business Intelligence platform XI 3.x or 4.0. The Data Services XI 3.x or 4.0 Metadata Integrator can continue to work with the Data Services XI 4.1 repository; however, this will not work with an SAP HANA repository.

You can continue to use the Data Services-specific impact and lineage. If you have a previous version, you can continue to view impact and lineage for objects from a Business Intelligence platform 3.x or 4.0 system after executing the previous version of Metadata Integrator to load metadata to the Data Services 4.1 repository.

For full functionality, you will need to use the Metadata/CMS integrator found in SAP BusinessObjects Information Steward.

Note:
You must add permissions for the Business Intelligence platform 4.0 FP3 user to retrieve the Data Services 4.1 repository password. Otherwise, their Metadata Integrator instance cannot get BusinessObjects object when running on Data Services 4.0. See the *Data Services Administrator's Guide* for more information.

6.2.2 Change in Global_DOP default value

The Global_DOP Job Server option default value has been changed from 1 to 2.

Related Topics

- USA Regulatory certification tests

6.2.3 Blank space padding and trimming

In this version of Data Services, a change was made that could affect output based on the values specified for the **Parallel process threads**, **Rows to read**, **Blank trimming**, and **Blank padding**

options. This update could result in changes to spacing behavior in your output. Verify the source and target file formats used in your jobs to ensure that the blank trimming and blank padding values reflect how the output should be generated.

6.2.4 USA Regulatory certification tests

This version of Data Services has changed several settings that can affect your existing certification jobs. The blueprints installed with this version have been updated; however, if you're using jobs based on blueprints from previous versions, you must update them.

When using the US Regulatory Address Cleanse or DSF2 Walk Sequencer transforms to process USPS CASS, DSF2, and NCOALink certification tests, you must have the following settings.

- The Global_DOP Job Server option default value has been changed from 1 to 2. However, for the US Regulatory Address Cleanse and DSF2 Walk Sequencer transforms, the value for the **Degree of parallelism** option must be set to 1 when processing a USPS certification test; otherwise, an error is generated. Because of this change to the default Global_DOP value, make sure that the jobs you are using for certification testing have a **Degree of parallelism** value of 1. Verify this setting in the data flow properties.
- For both the source and target file formats, the **Parallel Process Threads** option should be set to **{none}**. To do this, delete the existing value in the properties in the source and target workspace editor.
- For the source file formats, the **Blank Trimming** option should be set to **None**. Verify this setting in the source file format editor in the local object library.

6.2.5 Data Cleanse changes

This section describes the behavior changes in this release for the Data Cleanse transform.

Note:
If your environment includes customized Data Cleanse dictionaries, you need to upgrade to Data Services 4.1 and Information Steward 4.1 at the same time. The Data Cleanse transform in SAP BusinessObjects Data Services 4.1 relies on the availability of the Cleansing Package Builder functionality in SAP Business Objects Information Steward to upgrade custom Data Cleanse dictionaries or modified Data Services 4.0 person and firm dictionaries.

Engine option removed

The Japan engine option is no longer required for processing Japanese data and has been removed.

New options

The following new options have been added to the Options tab under Cleansing Package.

- Content Domain Sequence

- Output Format

Cleansing package structure has changed

Underlying changes to the cleansing package structure require that all cleansing packages be upgraded to the 4.1 version. Data Cleanse jobs will fail to execute if the cleansing package is version 4.0 or lower. This also applies to SAP-supplied person and firm cleansing packages and cleansing packages created or modified using Cleansing Package Builder.

Unmodified SAP-supplied person and firm cleansing packages: Download and install the version 4.1 SAP-supplied person and firm cleansing package. The version 4.1 cleansing package provides full support for domains (locales) and allows you to use the new Domain Content Sequence and Output Format options in the Data Cleanse transform.

To use a previously installed person and firm cleansing package, use the Cleansing Package Builder module of Information Steward to upgrade and then publish the cleansing package. NOTE: Support for domains (locales) in upgraded cleansing packages is limited to a single, global domain. Content from Data Services 3.2 person and firm dictionaries may be imported and merged with a newly created or SAP-supplied version 4.1 cleansing package to achieve full support for domains.

Modified SAP-supplied person and firm cleansing packages or custom cleansing packages created in Cleansing Package Builder: Use the Cleansing Package Builder module of Information Steward to upgrade and then publish the cleansing package.

Note:
Support for domains (locales) in upgraded cleansing packages is limited to a single, global domain. Content from Data Services 3.2 person and firm dictionaries may be imported and merged with a newly created or SAP-supplied version 4.1 cleansing package to achieve full support for domains.

New person and firm cleansing package

SAP has replaced the 20+ previously SAP-supplied person and firm cleansing packages with a new one that supports all locales in a single person and firm cleansing package. The new SAP-supplied person and firm cleansing package is now available for you to install and use. The new cleansing package includes all the data that existed in all previously shipped cleansing packages. When using the new cleansing package, you may notice some parsing differences. To use a previously-installed PERSON_FIRM_XX cleansing package, you need to go through the Cleansing Package Builder's cleansing package upgrade process, which will allow you to use the cleansing package as it worked in Data Services 4.0.

Cleansing package name option upgraded

Existing cleansing packages can be upgraded using the Cleansing Package Builder module of Information Steward. During the upgrade process, the cleansing package name is analyzed to determine if it is a known SAP-supplied cleansing package.

During the upgrade process, the Cleansing Package Name will be analyzed to determine if it is a known SAP-supplied cleansing package.

If the name of the cleansing package refers to an SAP-supplied cleansing package, the Cleansing Package Name option will be set to PERSON_FIRM and the Output Format and Content Domain Sequence options will be set as follows:

Cleansing Package	Original Cleansing Package Name	Content Domain Sequence setting	Output Format setting
Arabic	PERSON_FIRM_AR	AR|GLOBAL	AR
Benelux	PERSON_FIRM_BNL	NL|FR|GLOBAL	NL
Swiss	PERSON_FIRM_CH	DE|FR|IT|GLOBAL	DE
Czech/Slovak	PERSON_FIRM_CZ	CS|GLOBAL	CS
Danish	PERSON_FIRM_DA	DA|GLOBAL	DA
German	PERSON_FIRM_DE	DE|GLOBAL	DE
English	PERSON_FIRM_EN	EN_US|GLOBAL	EN_US
Spanish	PERSON_FIRM_ES	ES_MX|GLOBAL	ES_MX
French	PERSON_FIRM_FR	FR|GLOBAL	FR
Hungarian	PERSON_FIRM_HU	HU|GLOBAL	HU
Indonesian	PERSON_FIRM_ID	ID|GLOBAL	ID
India	PERSON_FIRM_IN	EN_IN|GLOBAL	EN_IN
Italian	PERSON_FIRM_IT	IT|GLOBAL	IT
Japanese	PERSON_FIRM_JP	JA|GLOBAL	JA
Malaysia/Singapore	PERSON_FIRM_MY	MS|GLOBAL	MS
Dutch	PERSON_FIRM_NL	NL|GLOBAL	NL
Norwegian	PERSON_FIRM_NO	NO|GLOBAL	NO
Polish	PERSON_FIRM_PL	PL|GLOBAL	PL
Portuguese	PERSON_FIRM_PT	PT_BR|GLOBAL	PT_BR
Romanian	PERSON_FIRM_RO	RO|GLOBAL	RO
Russian	PERSON_FIRM_RU	RU|GLOBAL	RU
Swedish	PERSON_FIRM_SE	SV|GLOBAL	SV

If the Cleansing Package Name does not refer to a SAP-supplied cleansing package:

- The Cleansing Package Name option will keep its current value.
- The Output Format option will be set to the name of the cleansing package.
- The Content Domain Sequence option will be set to **GLOBAL**.

Updates to Date options

The Data Cleanse Date options include the following updates.

Option group	Change	Current Description
Century Threshold	The default value has changed from 15 to 25.	Indicates whether a two-digit date is considered part of the 20th or 21st century. The default value is 25. Specify a two-digit integer that represents the first year that a parsed two-digit year is considered part of the 21st century (20xx). All two-digit years greater than the specified integer are considered part of the 20th century (19xx). For example, if you enter 11, all two-digit years 11 or lower are considered part of the 21st century. 08 is considered 2008. 11 is considered 2011. All two-digit years higher than 11 are considered part of the 20th century. 12 is considered 1912.

Option group	Change	Current Description
Date Delimiter	The value,**CHINESE_JAPANESE** is now available to support processing of Chinese and Japanese data.	Specifies what character to use for standard date output delimiters. **Backslash (\)**: Uses backward slashes as the delimiter for the date. For example, 04\01\2010. **Dash (-)**: Uses dashes as the delimiter for the date. For example, 04-01-2010. **Slash (/)**: Uses forward slashes as the delimiter for the date. For example, 04/01/2010. **None**: Does not add a delimiter to the date. For example, 04012010 **Period (.)**: Uses periods as the delimiter for the date. For example, 04.01.2010. **Space**: Uses spaces as the delimiter for the date. For example, 04 01 2010. **CHINESE_JAPANESE**: Uses the following Chinese/Japanese characters as delimiters: • 月 always follows the month • 日 always follows the day • 年 always follows the year An example of Arabic numbers with Chinese/Japanese delimiters is: 04月01日2010年 An example of Chinese/Japanese Numbers with Chinese/Japanese delimiters is: 四 月 一 日 二千零一 十 年
Numeric Format	This new option is added to set the numeric date values for Arabic or Chinese/Japanese numbers.	Specifies the format of numeric date values. **ARABIC_NUMBERS**: Returns numeric date values in Arabic **CHINESE_JAPANESE_NUMBERS**: Returns numeric date values in Chinese or Japanese.

Social Security Number file

The U.S. Social Security Administration is no longer updating its Social Security Number algorithm. Thus SAP no longer provides an updated DRLSSN.dat file. The file dated July 5, 2011 is the final file that SAP will provide.

For more information about the Social Security Administration's new assignment process, see http://www.ssa.gov/employer/randomizationfaqs.html .

6.2.6 Global Address Cleanse transform changes

This section describes the behavior changes in this release for the Global Address Cleanse transform.

Input field removed

The SUGGESTION_START_SELECTION input field is no longer required for processing and has been removed. If you currently have SUGGESTION_START_SELECTION mapped to your input schema, you need to delete it before you run the process again.

Standardization option value name changes

The following The Secondary Number Style option value names have been changed to more accurately describe the options.

- **Dashed** has been changed to **Attached**.
- **Trailing** has been changed to **Unattached**.

6.2.7 Change in Parallel process threads default value

The default value for the **Parallel process threads** option has been changed from 1 to 4. This change affects the File Reader, File Loader, and the SAP HANA Bulk Loader.

Impact to the US Regulatory Address Cleanse and DFS2 Walk Sequencer transforms

When using the US Regulatory Address Cleanse or DSF2 Walk Sequencer transforms to process USPS certification tests, the value for the **Parallel process threads** option must be 1 or an error will be generated. Because of this change to the default value, be sure that the jobs you are using for certification testing have a **Parallel process threads** value of 1.

6.2.8 Password protection for Data Services' repositories

When you log in to the Data ServicesDesigner or try to view a Data Quality report in the Management Console, you are now prompted to provide a user name and password for the Data Services repository.

See the *Data Services Administrator Guide* for more information, including information about how to change this default behavior.

6.2.9 User Access Control (UAC) support

Windows Vista and newer versions of Windows include a security feature known as User Access Control (UAC). In general, UAC addresses issues in system stability and security by running most applications with standard user rights, instead of administrator-level rights, as in previous versions of Windows.

In this version of SAP BusinessObjects Data Services, UAC support has been improved so that standard users do not require elevated access rights in order to run Data Services applications such as the Designer.

Configuration options

Previous versions of SAP BusinessObjects Data Services stored all configuration options in a location that required elevated access rights to modify. Because of this, standard users could run applications such as the Designer only in elevated access mode.

The Data Services installation program now creates additional environment variables: `DS_COMMON_DIR` on all platforms, and `DS_USER_DIR` on Windows platforms.

User-specific configuration options have been moved to a new file in `DS_USER_DIR`, where standard users have full access by default. This allows a standard user to run applications such as the Designer without requiring elevated access.

Configuration options	Old location	New location
Common options	<*LINK_DIR*>/bin/DSConfig.txt	<*DS_COMMON_DIR*>/conf/DSConfig.txt
User-specific options	<*LINK_DIR*>/bin/DSConfig.txt	<*DS_USER_DIR*>/conf/DSUserConfig.txt

Management Console and Adapters

In addition, several files used by the Management Console and adapters have been moved to the new `DS_COMMON_DIR` location.

File	Old location	New location
Management Console: admin.xml	<LINK_DIR>/bin	<DS_COMMON_DIR>/conf
Management Console: admin.key	<LINK_DIR>/bin	<DS_COMMON_DIR>/conf
Adapter: startup_script.xml	<LINK_DIR>/Adapters	<DS_COMMON_DIR>/adapters
Adapter XML files	<LINK_DIR>/Adapters/Config	<DS_COMMON_DIR>/adapters/config
Adapter: Adapter.key	<LINK_DIR>/Adapters	<DS_COMMON_DIR>/adapters

Job history, scheduled jobs, and logs

The batch files generated for scheduled jobs and exported execution commands contain a reference to the log folder location. In this version of Data Services, the location for log files has moved from <LINK_DIR>/log to <DS_COMMON_DIR>/log.

Note:
Old log files are not moved from their existing location in <LINK_DIR>/log. When a new or existing job is executed after Data Services has been upgraded to version 4.1, the new log file is be created in the <DS_COMMON_DIR>/log location.

Upgrading from previous versions

When you upgrade from Data Services 4.0.x or older by specifying an existing DSConfig.txt, the installation program upgrades the configuration options to the current version, and copies the common and user-specific configurations to <LINK_DIR>/conf/migrated.

When a user accesses the Designer on the upgraded installation, the Designer first looks for configuration information in <DS_USER_DIR>/conf. If the configuration information is not present, it copies the migrated configuration from <LINK_DIR>/conf/migrated to <DS_USER_DIR>/conf.

6.2.10 Changes to default rights for some user groups

Some user groups have had default rights changed.

In this version, the Data ServicesDesigner User group is by default granted the following rights:

- Execute Batch Job
- View Batch Job History
- Manage Batch Job History
- View Server Group Information
- Manage Server Group Configurations

In addition, when a local repository is registered using the Central Management Console, the following behaviors are different in this version:

- The Data Services Profiler Users are given No Access on a Data Services local repository by default.
- The Data Services Profiler Administrator Users are given No Access on a Data Services local repository by default.
- The Data Services Operator Users are given Full Control Access on a Data Services local repository by default.

6.2.11 Database pushdown with lookup_ext()

The lookup_ext function supports additional database pushdown optimizations.

Optimizing database push-down

For best performance, the lookup_ext function can be pushed down to the database when the following conditions are met:

- The lookup_ext function is used in the column mapping, output schema, or `SELECT WHERE` clause of a Query transform.
- The *`lookup_table`* is a database table from the same datastore or a linked datastore as the reader.
- The *`cache_spec`* is set to `NO_CACHE`.
- The *`return_policy`* is set to either `MAX` or `MIN`.
- All conditions used in the *`condition_list`* are database expressions.
- Only the equals operator (=) are used in the lookup *`condition_list`*.
- The *`run_as_separate_process`* SET option is set to `no`.
- For lookups with multiple-result column values, the database must support the rank (or equivalent) function.

Note:
For SAP HANA, MySQL, Sybase ASE, and Informix databases, no analytic function support is available. As a result, push-down is supported in all cases for single-result columns, and multiple-result columns only for primary keys.

For additional information about lookup_ext pushdown optimization and limitations, see the *Reference Guide*.

Related Topics

- Reference Guide: Functions and Procedures, Descriptions of built-in functions, lookup_ext

6.2.12 Monitor log changes

The monitor log sample rate is now based on a fixed timer instead of the number of rows processed. The new default value for the monitor sample rate is 30 seconds.

In addition, new performance data is written to the monitor log:

- The CPU utilization for each transform thread
- The number of rows in the input buffer and its used capacity (as a percentage)
- The number of input rows currently cached by each transform

Related Topics

- Administrator's Guide: Monitoring, Monitoring jobs
- Designer Guide: Executing jobs, Debugging execution errors, Using logs
- Performance Optimization Guide: Measuring Performance, Measuring performance of jobs, Reading the Monitor Log for execution statistics

6.2.13 Cleansing packages

Beginning in version 4.1, the SAP-supplied PERSON_FIRM_* cleansing packages are configured to handle all domains. This is now referred to as the global cleansing package. You can merge from PERSON_FIRM_* cleansing packages that were locale-specific into the single, global cleansing package.

Note:
The individual SAP-supplied cleansing packages (PERSON_FIRM_*) will no longer have their content/rules updated.

The Data Cleanse transform in SAP BusinessObjects Data Services 4.1 relies on the availability of the Cleansing Package Builder functionality in SAP Business Objects Information Steward to upgrade previous versions of custom Universal Data Cleanse dictionaries or modified SAP-supplied cleansing packages or modified Data Services XI 3.2 person and firm dictionaries to the current version.

Existing Data Services XI 3.2 dictionaries can be imported or merged and used in Cleansing Package Builder 4.1. Data Services/Information Steward 4.0 cleansing packages can also be used in Data Services 4.1, but you must first open the cleansing packages in the 4.1 version of Cleansing Package Builder to update the schema. However, only cleansing packages that have been merged with the global cleansing package will have updated content and rules in future releases. Previous versions of cleansing packages that begin with PERSON_FIRM_* will not be updated going forward. See the *Information Steward Upgrade Guide* for information about merging previous versions to the current version.

6.2.14 JDBC driver handling for Oracle and MySQL

Data Services no longer bundles JDBC drivers for Oracle and MySQL databases. If you use an Oracle or MySQL database for your Data Services repository, you must download the JDBC driver from your database vendor.

During the installation or upgrade process, the Data Services installation program will ask you to provide the location of the JDBC driver for your database. You can also choose to configure the JDBC driver after installation by manually copying files to the locations required by Data Services.

For more information, see “Post-installation, Configuring JDBC drivers for SAP HANA, Oracle, and MySQL” in the *SAP BusinessObjects Data Services Installation Guide*.

6.2.15 SAP ABAP function updates and new namespace

To take advantage of the new functions and /BODS/ namespace changes, delete all functions that were installed with previous versions and all related ZAW* objects.

To install the new functions, see the *Supplement for SAP: Installing Functions on the SAP Server*.

6.2.16 RFC streaming

If you are already using ABAP data flows that use the **Execute Preloaded datastore** option, you must regenerate the ABAP code and upload the generated programs to update the SAP system. To do so, in the object library right-click each ABAP data flow name and select **Generate ABAP code**. In the dialog box, select **Upload generated programs**.

6.3 Behavior changes in version 14.0.0

The following sections describe changes in the behavior of SAP BusinessObjects Data Services XI 4.0 (14.0.0) from previous releases of Data Services and Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see Data Quality to Data Services Upgrade.

This section includes migration-specific information associated with the following features:

- Management Console
- Designer
- 64-bit on Windows and Linux
- Central repository users and groups
- Importing and exporting objects
- SSL protected communication channels
- BusinessObjects Enterprise Metadata Integrator

- Microsoft Excel
- Joins in the Query transform
- Unsupported BWA datastore database type
- SAP NetWeaver RFC library
- ODBC usability on UNIX
- Auto Documentation image display
- Data Quality support for NULL values
- Data Quality support for native data types
- Data Cleanse solution
- Global Address Cleanse transform
- Global Suggestion List transform
- Match transform
- USA Regulatory Address Cleanse transform
- GeoCensus and Geocoder

6.3.1 Management Console

With this release, some administrative functions previously available in the Administrator have been moved to the Central Management Console (CMC):

- User and group management
- Repository registration
- Log retention configuration

The following items have been removed from the Management node of the Administrator navigation tree:

- Users
- Repositories
- Log retention period

For more information about user management, see the *Administrator's Guide*.

Logging into the Management Console

In previous versions, user access to the Management Console was controlled by defining users in the Administrator. With this release, access to the Management Console is configured in the Central Management Console (CMC).

Management Console access can be granted to users or groups defined in the CMC. Users are required to log into the Management Console using these credentials. Management Console applications to which the user has not been granted access are not available.

For more information about logging into the Management Console, see the *Management Console Guide*.

6.3.2 Designer

In previous versions, no user authorization was required to access the Designer application, and repository access was controlled directly by the repository database connection. With this release, repository connections are configured in the Central Management Console (CMC).

Repository access can be granted to users or groups defined in the CMC. Users are required to log into the Designer using these credentials. When logging in, the repositories to which the user has access are displayed.

Note:
Connecting to a central repository from within the Designer now requires the same method of authorization.

For more information about user management, see the *Administrator's Guide*.

64-bit Designer

A native 64-bit Designer is now available in addition to the 32-bit Designer.

There are some new requirements and restrictions on 64-bit, such as Universe Builder and Microsoft Excel functionality. For more information, see the *Release Notes*.

6.3.3 64-bit on Windows and Linux

In this release of SAP BusinessObjects Data Services, a 64-bit job server is available on Windows and Linux, and a 64-bit Designer is available on Windows. This means that 64-bit middleware (for databases and applications) is now required. 32-bit middleware drivers cannot be used.

Teradata interface on Windows 64-bit platform

In order to use the Teradata interface on a Windows 64-bit platform, you must set the following environment variable on the Windows job server machine:

```
LIB_32BIT_TERADATA=<Teradata 32-bit client location>
```

This step is necessary due to the unavailability of Teradata client utilities on Windows 64-bit platforms.

6.3.4 Central repository users and groups

In previous versions, users and groups for central repositories were managed in the Management Console Administrator application. With this release, central repository users are now standard users defined in the Central Management Server (CMS).

Central repository groups are still managed in the Administrator.

For more infromation about user management, see the *Administrator's Guide*.

6.3.5 Importing and exporting objects

In previous versions, when you exported repository objects to a file, any passwords stored in the objects were encrypted using an internal key. With this release, you are asked for a passphrase that is used for encryption when you export repository objects to a file. If you do not specify a passphrase, any passwords stored in the objects are removed.

When you import objects from a file, you are asked for the encryption passphrase. If you do not enter the passphrase, or the passphrase does not match the one used to export the file, any passwords stored in the objects are removed and must be manually entered after the import process has finished.

Note:
If you import a file from a previous version of Data Services, any passwords for the imported objects are empty, and need to be entered after the import process has finished.

For more information about importing and exporting objects, see the *Designer Guide*.

6.3.6 SSL protected communication channels

By default, the communcation channels between Data Services components are now protected by SSL. The SSL encryption may result in a performance impact to some jobs that are run as a separate process. If SSL protection is not required, you can disable it, and there will be no SSL-related performance impact compared to previous versions.

In addition, any message client applications need to be re-linked with the new library to take advantage of the SSL protection. Applications may continue to use the existing library if SSL is disabled for the Access Server.

For more information about configuring SSL, see the *Administrator's Guide*.

6.3.7 BusinessObjects Enterprise Metadata Integrator

If you are migrating from BusinessObjects Data Services XI 3.x, and have BusinessObjects Enterprise Metadata Integrator configured to collect from a CMS repository version XI 3.x, you must retain the BusinessObjects Enterprise 3.x client machine with Data Services XI 3.x Metadata Integrator installation to continue to collect CMS metadata from BusinessObjects Enterprise XI 3.x systems. From Data Services XI 4.0, you cannot directly configure Metadata Integrator to collect from BusinessObjects Enterprise XI 3.x. The Data Services XI 3.x Metadata Integrator can continue to work with the Data Services XI 4.0 repository.

6.3.8 Microsoft Excel

The Microsoft Excel interface has software dependencies in SAP BusinessObjects Data Services 4.0 and later. Special care must be taken when installing Data Services on a machine with a Microsoft Office version lower than 2010. For details, see the *Installation Guide*.

6.3.9 Joins in the Query transform

Data Services now supports ANSI/SQL92 joins in the Query transform. In the Query editor, the OUTER JOIN tab has been removed and the functionality in the FROM tab has been extended to enable you to specify all join types in one location and to more accurately define restrictions. Additionally, join rank and cache can now be set directly in the FROM tab of the Query editor.

During upgrade

During upgrade, the following actions occur:

- The Query transform ATL and XML are automatically upgraded during the repository upgrade.
- Inner join queries are not affected. Inner joins previously specified in the WHERE clause will remain the same.
- Left outer joins are moved from the OUTER JOIN tab to the FROM tab.
- Predicates in the WHERE clause may be moved to the FROM clause. To resolve any ambiguity of filter conditions for left outer joins, the Data Services engine takes the following actions:
  - When a search condition is defined on the outermost table of a left outer join, the search condition is considered a filter and placed in the WHERE clause of the SELECT statement.
  - When a search condition is defined on an inner table of a left outer join, the search condition is considered a condition of the left outer join and is placed in the FROM clause of the SELECT statement.

After upgrading

During upgrade data flow definitions are validated. Valid data flows upgrade seamlessly. Errors may occur for invalid data flows. After upgrading Data Services, review the error log and trace log files for

any errors which occurred during upgrade. As necessary, manually edit the data flows and query transforms where errors occurred.

The logs are located in the <*LINK_DIR*>\admin\repo directory.

Join rank and cache settings

In migrated jobs, join ranks are inherited from the sources and there should be no change in join order.

After upgrade, the best practice is to set join rank and cache settings in the FROM tab of the Query editor. Join Rank and cache settings set at the source level are supported. However, join rank or cache settings in the Query editor override settings in the source. If a default join rank setting in the Query editor is changed, all join ranks set in the sources for that query are overridden. Similarly, if a default cache setting in the Query editor is changed, all cache settings set in the sources for that query are overridden.

6.3.10 Unsupported BWA datastore database type

The SAP NetWeaver Business Warehouse Accelerator (BWA) datastore is no longer supported in SAP BusinessObjects Data Services XI 4.0, as well as its related functions: begin_trex_transaction, commit_trex_transaction, create_trex_cube, delete_trex_cube, and rollback_trex_transaction.

If you need the functionality to load to BWA, do not upgrade to SAP BusinessObjects Data Services XI 4.0 or later.

6.3.11 SAP NetWeaver RFC library

Data Services now uses the SAP NetWeaver RFC SDK library to communicate with SAP systems, replacing the SAP Classic RFC ASCII library. Please note the following upgrade issues:

- Data Services no longer supports the IDoc release 3.x format. In SAP, upgrade any IDocs used in existing Data Services jobs to version IDoc 4.x.
- If you are using an `saprfc.ini` file, the new library requires the file name to be `sapnwrfc.ini`, and you must update the parameters in the file to follow the syntax and semantics documented in *SAP NetWeaver RFC SDK* for version 7.1.
- Data Services no longer supports the command-line RFC Server interface. If you previously used the command-line interface, configure all RFC Server instances in the Management Console Administrator.
- To enable loading data to SAP NetWeaver BW targets from multi-byte code pages, upgrade your SAP system to Unicode. For more information, refer to SAP Note 838402, "Problems with non-Unicode system landscapes," at https://service.sap.com/sap/support/notes/838402.

- If you changed your SAPCODEPAGE environment variable, by default SAP NetWeaver RFC accepts logon data in SAPCODEPAGE=1100. If you configured a datastore for any other character set, reset the appropriate SAPCODEPAGE environment variable.
- If you connect to a non-Unicode SAP system, set the datastore "Locale" options appropriately to process character sets in a specific language. For example, to process Japanese data in a non-Unicode SAP system, set the datastore **Language** option to **J**. Set the SAP datastore **Code page** in the following situations:
  - For ABAP data flow processing in SAP R/3 4.x systems.
  - For processing IDoc files.
- If you use the Data Services**Direct download** data transport method and ABAP upload feature, ensure that you have implemented the support packages described in SAP Note 1258724, "Starting SAPGUI using external RFC libraries." With the upgrade from the classic RFC library to the NetWeaver RFC library, these support packages provide two modules SYSTEM_PREPARE_ATTACH_GUI and SYSTEM_FINISH_ATTACH_GUI, which are required for SAPGUI support. The note is available at https://service.sap.com/sap/support/notes/1258724.
- If you use SAP R/3 4.6C and the SAP datastore user name and password are not in uppercase, implement the solution documented in SAP Note 792850, "Preparing ABAP systems to deal with incompatible passwords." The note is available at https://service.sap.com/sap/support/notes/792850.

For more information, see the *Supplement for SAP*.

6.3.12 ODBC usability on UNIX

This version of Data Services includes several changes that improve the configuration of ODBC data sources on UNIX platforms:

- In previous versions, Data Services used an ODBC configuration file called `odbc.ini`. This file has been renamed `ds_odbc.ini` to reduce confusion related to the `odbc.ini` file used by many ODBC driver managers.

  In addition, the configuration file is no longer required to configure natively-supported ODBC data sources.

- Data Services now includes a utility that simplifies configuration of natively-supported ODBC data sources. By default, this utility is installed to `<LINK_DIR>/bin/dsdb_setup.sh`.

  ODBC data sources natively supported by Data Services include:
  - HANA
  - MySQL
  - Neoview
  - Netezza
  - Teradata

  For other ODBC data sources, additional manual configuration is required.

For more information, see “Configuring ODBC data sources on UNIX” in the *Administrator's Guide*.

Migrating existing configurations

With this version, ODBC configurations are stored in the `ds_odbc.ini` file, which is located in the `<LINK_DIR>/bin` directory. In previous versions, this file was named `odbc.ini`.

To migrate your existing configurations, copy the contents of your Data Services XI 3.x `odbc.ini` into the new `ds_odbc.ini` file.

Alternatively, you can configure new ODBC data sources with the `dsdb_setup.sh` script after installation.

For more information, see “Configuring ODBC data sources on UNIX” in the *Administrator's Guide*.

6.3.13 Auto Documentation image display

In the Management Console Auto Documentation feature, large images greater than a specific size can now be displayed in a partial view to improve performance and memory utilization. A new **Image Display** option has been added that lets you specify whether a whole or partial image is displayed.

To access the **Image Display** option, click **Settings** in the upper-right corner of the Auto Documentation page in your browser. By default, the option is set to Partial.

6.3.14 Data Quality support for NULL values

To utilize the Data Quality support for NULL in your current jobs, manually make some adjustments to your existing jobs after upgrade.

If you want to keep the way NULL fields are processed in the previous version of the software, you don't have to make any adjustments to your upgraded jobs.

If you want to add the support for NULL in your existing jobs, follow these steps:

1. Right-click on the "Dataflow" icon to open the "Dataflow Properties" window.
2. Open the "Attributes" tab.
3. Select Yes for the new option **ANSI_varchar_DQ**.
4. Click **Apply** and then **OK** to close the "Properties" window.

Related Topics

- Designer Guide: Platform support for NULL values

6.3.15 Data Quality support for native data types

New field attributes: When upgrading existing jobs that contain Data Quality transforms to the new version of the software, the software issues a general information message stating that the software added Type, Length, and Scale to all Data Quality input and output fields. Open your existing jobs to see the new attributes.

New data types: When upgrading jobs that contain fields with new data types, the converted data type is applied to the fields listed in the Output tab (lower portion). However, conversion does not affect the Schema Out setting of your existing jobs, so your jobs will still run. After you upgrade, the downstream transforms will still see the same data types streaming out.

If you want to take advantage of the data type change in your existing jobs, you can change the Schema Out data type to match the field's new data type.

Find out if the Data Quality transforms that you use have fields with new data types by reading the *What's New* document.

6.3.16 Data Cleanse solution

Changes to structure require that you upgrade all Data Services 3.2 Data Cleanse dictionaries to cleansing packages of the current version. In order to execute successfully, a Data Cleanse job must reference a cleansing package. The cleansing package may be either an SAP-supplied cleansing package or a cleansing package that you have modified and published in the Cleansing Package Builder module of Information Steward.

Restriction:
If you currently have modified a person and firm dictionary or created a custom dictionary using Universal Data Cleanse, ensure that the Cleansing Package Builder in Information Steward is available before you upgrade. You will need Cleansing Package Builder to migrate your dictionary rules, and reference files to the new cleansing package format.

6.3.16.1 Prior to upgrading

Upgrading unchanged SAP-supplied cleansing packages

If you are using SAP-supplied Data Services 3.2 cleansing packages (dictionary, rules, and reference files) that have not been modified, no special action is required before the upgrade.

Before choosing to upgrade, keep in mind that the SAP-supplied cleansing packages are enhanced for each release and you may notice some changes in your cleansed data. Additionally, after upgrading, to update the Social Security number file, you must use Cleansing Package Builder. If regular updates to Social Security number information are critical, ensure that Cleansing Package Builder is available prior to upgrading Data Services.

After upgrading Data Services, you must use the Cleansing Package Installer to install the current SAP-supplied cleansing packages.

The new SAP-supplied cleansing packages follow the same naming convention previously used for the **Parsing Dictionary** option. In upgraded Data Cleanse jobs, the **Cleansing Package** option defaults to the name previously used in the **Parsing Dictionary** option.

Upgrading modified SAP-supplied or custom (Universal Data Cleanse) dictionaries, rules, and reference files

If you have modified your Person and Firm dictionary or have created a custom (Universal Data Cleanse) dictionary, before uninstalling Data Services 3.2 you must do the following:

- Ensure that Cleansing Package Builder in Information Steward is available to you.
- Export all dictionary changes (**Dictionary > Export Dictionary Changes**).
- Archive the reference and rule files. By default, the files are located in the `<LINK_DIR>\DataQuality\datacleanse` directory and have the following names:

File type	Default file name
Parsing rules	`drlrules.dat` for English; other regional rule files follow the convention `dc_rules_person_firm_<language>.dat`. There is not a default name for custom Universal Data Cleanse files.
Email domain names	drlemail.dat
International phone number patterns	drlphint.dat
User defined pattern matching (UDPM)	drludpm.dat

Note:
It is not necessary to archive the Social Security numbers file, drlssn.dat. You should update this information within Cleansing Package Builder immediately prior to publishing a cleansing package.

Use the Cleansing Package Builder in Information Steward to create a cleansing package by importing the archived Data Services 3.2 dictionary and files.

Note:
The archived Data Services dictionary and files must be from Data Services version 3.2. If you have a prior version of Data Services and wish to upgrade Data Cleanse files, you must upgrade to Data Services 4.0 before upgrading to Data Services 4.0.

6.3.16.2 Data Cleanse changes

Dictionary menu

The Cleansing Package Builder module of Information Steward is required in order to modify or customize any type of data. The dictionary menu has been removed from the Data ServicesDesigner menu bar.

Cleansing package repository

Data Cleanse no longer requires a separate cleansing package repository.

Changes in transform options

With the introduction of the Cleansing Package Builder in Information Steward you no longer need to specify individual dictionary, rules, and reference data files. The information formerly contained in those files is now included in the cleansing package.

The Cleansing Package option group and Cleansing Package option has been added to the Data Cleanse transform.

The following options are no longer available in the transform:

Option Group	Option Name
Dictionary	Parsing Dictionary
Reference Files	Rule File
	Email File
	International Phone File
	Social Security File
	User Defined Pattern File

Additionally, the name of the **Output Text Width Conversion** option has been changed to **Character Width Style**. The behavior of the option remains the same.

Change in capitalization for mixed strings

The default standardized output for strings that contain either both alphabetic and numeric characters or both alphabetic and punctuation characters has changed. The following table describes the new behavior:

Mixed string	Data Services 14.0.0 behavior	Input	Data Services 12.2.x output	Data Services 14.0.0 output
letters following punctuation	letters are lower case	ALL_BY_NET	All_By_Net	All_by_net
letters following numbers	letters are upper case	7MILES	7miles	7Miles
letters between numbers	letters are lower case	4X4	4X4	4x4

Note:
To change the default behavior, use Cleansing Package Builder to define the desired standard form for each affected variation.

6.3.16.3 After upgrading

You must run the Cleansing Package Installer to install SAP-supplied cleansing packages.

Prior to executing a job which contains a Data Cleanse transform, ensure that the desired cleansing package is selected in the Cleansing Package option of the transform.

Log files

During the upgrade, changes are made to the Data Cleanse transform that affect the behavior of existing jobs. Log files detailing the changes are available in the `<LINK_DIR>\Admin\Repo` and `<LINK_DIR>\log` folders.

6.3.17 Global Address Cleanse transform

This section describes the behavior changes in this release for the Global Address Cleanse transform.

Modifications to increase usability and deployability

As part of this release, the Global Address Cleanse transform underwent modifications to increase usability and deployability. These modifications involved the consolidation of engines and options within the transform along with the elimination of the out of process architecture that existed in the version 3.x releases. In making these changes, some address assignment discrepancies were introduced when compared side-by-side with prior versions. This is especially true for certain EMEA regions. These

discrepancies are being tracked and evaluated, and with continual efforts being made to improve and tune address assignment functionality, will be pushed into future version 4.x releases.

Country engines removed

The Australia, EMEA, and Japan engine option groups have been removed from the Global Address Cleanse transform. Some options were retained and moved to the Global Address engine option groups. The table lists the option names that are still in use and new location.

If you are using the Australia, EMEA, or Japan engines, in current dataflows, run the repository manager to update your current data flows. You may need to update the reference file paths and variables.

<table>
<tr><th>Option(s)</th><th>Located in</th></tr>
<tr><td>Australia reports options
Customer Company Name
List Name
File Name
Mailer Address1
Mailer Address2
Mailer Address3
Mailer Address4</td><td>Global Address > Report Options > Australia</td></tr>
<tr><td>Australia engine > Disable Certification</td><td>Global Address > Country Options > Country > Options > Disable Certification</td></tr>
<tr><td>EMEA engine > Street Name Style</td><td>Standardization Options > Country > Options > Street Name Style (Netherlands only)
<table><tr><th>Old setting</th><th>New setting</th></tr><tr><td>Official</td><td>Preserve: Preserves the street data format as it was input.</td></tr><tr><td>NEN5828</td><td>Short: Outputs street data in the format preferred by the Ministry of Internal Affairs. Street address with maximum 24 characters in mixed case.</td></tr></table></td></tr>
<tr><td>Japan engine > Retain Postcode if Valid Format</td><td>Global Address > Country Options> Country > Options > Retain Postcode if Valid Format</td></tr>
</table>

Option(s)	Located in
Japan engine > Dual Address	Global Address > Country Options > Country > Options > Dual Address

Updates to components

The components, PRIMARY_NUMBER_LOW and PRIMARY_NUMBER_HIGH are removed from the Global Address Cleanse transform's list of output components as they are only used by the Suggestion Lists option. The Suggestion List components are located in the under the new Suggestion List option group on the Global Address Cleanse transform Options tab.

6.3.17.1 Address Server removed

The Address Server is no longer required by the Global Address Cleanse and Global Suggestion Lists transforms to process global addresses. The functionality provided by the Address Server is now available in-process within the Global Address Cleanse and Global Suggestion Lists transforms.

The installation and upgrade processes do not remove the Address Server. You need to manually stop the Address Server and delete the Address Server folder and files located on Windows in `<DataServicesInstallLocation>:\Business Objects\BusinessObjects Data Services\bin\` and on UNIX in `<DataServicesInstallLocation>:/Business Objects/BusinessObjects Data Services/bin/`.

6.3.17.2 Space requirements for international addressing directories

Due to additional country support and modified database structures (for performance tuning), the minimum and maximum disk space requirements for the international addressing directories (All World) are as follows:

- For the Global Address Cleanse transform, the minimum requirement 2 GB.
- If you purchase all countries, the disk space requirement has increased from 9.34 GB to 33 GB.

6.3.17.3 Native data types and NULL

For upgrade information, refer to the information about platform support for Native data types and NULL.

Related Topics

- Data Quality support for native data types
- Data Quality support for NULL values

6.3.17.4 New Suggestion Lists options group

The Global Address Cleanse transform's Global Address engine now supports the Suggestion List option. When the Suggestion List option is enabled, the Global Address Cleanse transform can return a list of assignment candidates to choose from.

The Suggestion List options group in the Global Address Cleanse transform has changed.

Restriction:
These changes make SAP BusinessObjects Data Services XI version 14.0.0.0 non-backward-compatible.

The list below contains the areas that have changed for suggestion lists:

- Options
- Output components
- XML output features

6.3.17.4.1 New and changed Suggestion List options

In the new version of the Global Address Cleanse transform, the Suggestion List Options group is renamed to Suggestion List and the hierarchy of the Suggestion List group is “flattened” so that the options are easier to view and to set. In addition, some of the option names have changed to more closely align with other transforms in Data Services.

Deleted options

Previously, the Suggestion List Options group contained a group named Output that was divided into Lastline Components, Primary Address Components, and Secondary Address Components. These options have been “flattened” in the new version of the transform for simplicity, and are located in a new group named Suggestion List Components, which replaces the Output group.

New option group: Suggestion List Components

The new Suggestion List Components group of options is added to the Suggestion List group. It contains the suggestion list address components. Previously the suggestion list address components were in the Suggestion List group. The Suggestion List Components option includes the 49 fields that you can choose to include in the Suggestion_List output field. For a complete list of fields, see the Reference Guide.

6.3.17.4.2 New XML output feature

When you choose to output the Suggestion_List field in XML format, you will notice tags that reflect the renaming of some of the address components. Here is a list of the new tags:

New tags	Pevious tags
<LOCALITY1>	<LOCALITY1_OFFICIAL>
<POSTCODE>	<POSTCODE1> <POSTCODE_FULL>
<PRIMARY_NAME_FULL1>	This is a new tag.
<ENTRY>	<LIST>

New suggestion list tag

There is also a new tag for suggestion list.

```
<SUGGESTION_LIST xmlns=”urn:bobjsap_gac”></SUGGESTION_LIST>
```

New XML/XSD output

Here is an example of the new suggestion list output for XML (sample has been reformatted for easier viewing):

```
<SUGGESTION_LIST xmlns=”urn:bobjsap_gac”>
<ENTRY><SELECTION>1</SELECTION>...</ENTRY><ENTRY><SELECTION>2</SELECTION...</ENTRY></SUGGESTION_LIST>
```

6.3.18 Global Suggestion List transform

The Global Address Cleanse transform's Global Address engine now supports the Suggestion List option. When the Suggestion List option is enabled, the Global Address Cleanse transform can return a list of assignment candidates to choose from. If you want to query the global address data and are performing wildcard searching, continue to use the Global Suggestion List transform.

Changes to Global Suggestion Lists transform engines

The following engines have been removed from the Global Suggestion List transform.

Option	New option
Australia engine	Global Address engine: Use this engine to query global address data for all countries other than the United States and Canada.
Multi Country engine	Global Address engine: Use this engine to query global address data for all countries other than the United States and Canada.

Update current dataflows

If you currently use the Global Suggestion Lists transform in a data flow used for global address correction, you need to remove it from the data flow and update the Global Address Cleanse transform's Suggestion Lists options.

Updated component name

The component, Actual_Assignment_Level has been renamed Assignment_Level. The component name will change during the upgrade. You do not need to make any changes to your current data flows.

6.3.19 Match transform

The following Match transform output fields' data type has changed from CHARACTER to INTEGER32:

- GROUP_NUMBER,
- MATCH_SCORE, PRIORITY_VALUE
- Group Stats GROUP_COUNT
- Group Stats GROUP_ORDER, SOURCE_COUNT
- SOURCE_ID_COUNT
- Input Source Stats GROUP_COUNT
- Input Source Stats GROUP_ORDER
- GROUP_SOURCE_APPEARANCE
- GROUP_SOURCE_ORDER
- GROUP_SOURCE_GROUP_APPEARANCE
- GROUP_SOURCE_GROUP_ORDER
- MULTI_SOURCE_COUNT
- Input Source Stats SOURCE_COUNT
- SOURCE_GROUP_COUNT.

You will see the change in data type in Match, but Data Services will return the data type you have defined for the field on output.

6.3.20 USA Regulatory Address Cleanse transform

This section describes the behavior changes in this release for the USA Regulatory Address Cleanse transform.

6.3.20.1 Changes to suggestion lists

The Suggestion List options group has changed. The new Suggestion List group and other field and option changes make SAP BusinessObjects Data Services XI version 14.0.0.0 non backward-compatible.

6.3.20.1.1 Changes to suggestion list output field components

Some of the output fields in the new Suggestion List Components group have changed. After you upgrade to the new version of the transform, the new fields are set as described in the following table.

Previous option name	New option name	Upgrade setting
Locality1 Locality1 Official	Locality1	Settings for Locality1 Official (lastline component) and/or Locality1 (primary address component) in your current jobs transfer to the Locality1 option. If your current job has one option set to Yes and the other set to the substitution variable, the Locality1 option is set to the substitution variable and the software issues a warning message. If your current job has one of the fields set to Yes and the other set to No, the Locality1 option is set to Yes.

Previous option name	New option name	Upgrade setting
Postcode1 Postcode Full	Postcode	Settings for Postcode1 (primary address component) and/or Postcode Full (lastline component) in your current jobs transfer to the new Postcode option. If your current job has one option set to Yes and the other set to the substitution variable, the new Postcode option is set to the substitution variable and the software issues a warning message. If your current job has one of the fields set to Yes and the other set to No, the new Postcode option is set to Yes.
N/A	Primary Name Full1	The new Primary Name Full1 option is added to your current jobs and is set to No by default. The Primary Name Full1 field contains the Primary Prefix1, Primary Name1, Primary Type1, and the Primary Postfix1 fields. However, these fields still exist in the new version of the transform and their Yes/No settings transfer to the new version. When setting up new jobs, keep in mind that you can set the Primary Name Full1 to Yes to include all four fields that have been combined into one component of the Suggestion_List output field.

6.3.20.1.2 New and changed options

The Suggestion List Options group is renamed to Suggestion List and the hierarchy of the Suggestion List group is “flattened” so that the options are easier to view and to set. In addition, some of the option names have changed to more closely align with other transforms in Data Services.

Deleted options

Previously, the Suggestion List Options group contained a group named Output that was divided into Lastline Components, Primary Address Components, and Secondary Address Components. For these options, the address designator (primary, secondary, lastline) have been removed for simplicity. The new group named Suggestion List Components, which replaces the Output group, contains the output fields that you choose for output.

New option group: Suggestion List Components

The new Suggestion List Components group of options is added to the Suggestion List group. It contains the suggestion list address components. Previously the suggestion list address components were in the Suggestion List group.

Options that have moved

The following options have been moved from **Suggestion List Options** > **Output** to the renamed Suggestion List group :

- Style
- Delimiter
- Field Delimiter

Related Topics

- Reference Guide: Transforms, USA Regulatory Address Cleanse, Suggestion list components

6.3.20.1.3 Output field levels

The table below lists the Suggestion List components and their applicable address levels.

Suggestion List component	Primary	Secondary	Lastline
Selection	X	X	X
Locality1	X	X	X
Region1	X	X	X
Postcode	X	X	X
Primary Number Low	X	X	
Primary Number High	X	X	
Primary Prefix1	X	X	
Primary Name1	X	X	
Primary Type1	X	X	
Primary Postfix1	X	X	
Primary Name Full1	X	X	
Postcode2 Odd	X	X	
Postcode2 Even	X	X	
Primary Side Indicator	X	X	
Firm		X	
Unit Description		X	
Unit Number Low		X	
Unit Number High		X	

Suggestion List component	Primary	Secondary	Lastline
Secondary Side Indicator		X	

Note:
If you choose XML for the output style, fields with blank values will not be output to the Suggestion_List output field. Previously, the field would be output with empty tags (no value).

6.3.20.1.4 New XML output features

When you choose to output the Suggestion_List field in XML format, you will notice root elements that reflect the renaming of some of the address components. Here is a list of the new root elements:

New element	Previous element
<LOCALITY1>	<LOCALITY1_OFFICIAL>
<POSTCODE>	<POSTCODE1> <POSTCODE_FULL>
<PRIMARY_NAME_FULL1>	This is a new tag.
<ENTRY>	<LIST>

New suggestion list element

There is also a new root element for suggestion list.

```
<SUGGESTION_LIST xmlns=”urn:bobjsap_urac"></SUGGESTION_LIST>
```

These root element changes enhance your ability to parse the suggestion list output results.

Example: New XML/XSD output

Here is an example of the new suggestion list output for XML (sample has been reformated for easier viewing):

```
<SUGGESTION_LIST xmlns=”urn:bobjsap_urac"> <ENTRY><SELECTION>1</SELEC
TION>...</ENTRY><ENTRY><SELECTION>2</SELECTION...</ENTRY></SUGGESTION_LIST>
```

6.3.20.2 Updates to support multiple data source statistics reporting

To support the new feature to generate statistical reports for multiple lists in an input database, there is a new option added to the Report and Analysis group named **Gather Statistics Per Data Source**.

If your current jobs include the input field Data_Source_ID in the input field mapping, and the **Generate Report Data** option is set to Yes, the new version of the transform sets the option **Gather Statistics Per Data Source** in the Report and Analysis group to Yes.

Related Topics

• Designer Guide: Data Quality, Beyond the basic address cleansing, Multiple data source statistics reporting

6.3.20.3 Updates to support USPS Certifications

When you upgrade to the new version of the transform, the following changes are made to support USPS certifications:

- The output field NCOALink_Stage_Record is replaced with a new field named Stage_Test_Record. If any existing jobs contain the NCOALink_Stage_Record field in output mapping, the field mapping will change to Stage_Test_Record.
- A new option named **USPS Certification Testing Mode** is added to the Assignment Options group with the values of CASS, DSF2 Augment, or None. The option is set to the default setting of None.

Related Topics

• Designer Guide: Data Quality, Beyond the basic address cleansing, USPS certifications

6.3.20.4 Change to report percentage calculation

With this release there is a change in how the USA Regulatory Address Cleanse transform calculates the denominator value used in percentage calculations. This change affects two Data Quality reports: US Addressing Report and the USA Regulatory Address Cleanse version of the Address Validation Summary report.

US Addressing Report: The calculation of the denominator is based on the total of all DPV return codes of Y, D, S, and the addresses that are not DPV valid.

Address Validation Summary report: The calculation of the denominator is based on the total of all records processed by the USA Regulatory Address Cleanse transform.

For both reports, the following record types will no longer be factored into the percentage calculation:

- Number of records where all input fields mapped into the transform contain NULL or blank values.
- Number of records that contain Unicode characters.
- Number of records that are not processed because of Z4Change functionality.

Previously, for both reports, the calculation for the percentages was based on using a denominator value of the total of all records processed by the job.

Related Topics

- Management Console Guide: Data Quality reports, US Addressing Report, Percent calculation

6.3.20.5 Native data types and NULL

For upgrade information, refer to the information about platform support for Native data types and NULL.

Related Topics

- Data Quality support for native data types
- Data Quality support for NULL values

6.3.20.6 Field size and data type changes

Data Quality field size changes

Because some of the field lengths have changed, you may receive warning messages that your input data will be truncated if it exceeds the stated field length.

The table below shows the field sizes that have changed in this release of the USA Regulatory Address Cleanse transform.

Field	Old field length	New field length
Address_Line	60	90
Locality2	60	61
Multiline(1-12)	60	80
Suggestion_List	30,000	60,000

Data Quality data type changes

The Suggestion_Start_Selection input field has changed from character to int data type. The Count and Error output fields have changed from character to int data types.

6.3.20.7 Verification warnings

The following situations may cause messages or warnings when you upgrade your existing jobs to the new version of the transform:

- An information message appears that informs you that the software has added Data_Type, Field_Length, and Field_Scale to input and output field attributes.
- A warning message appears if the number of characters in an input field exceeds the maximum size set in the new version of the USA Regulatory Address Cleanse transform. The message warns that your data will be truncated.
- When you upgrade your existing jobs, the software may update field lengths to the maximum and may log warning messages as applicable.

6.3.20.8 Other migration considerations

Changes in the software can result in jobs generating error messages that used to run through to completion in previous versions. To avoid errors, check your current jobs before you upgrade to the new version of the software.

Report generation and CASS certification

When your current job has **Disable Certification** in the Non Certified Options group set to No, you must also set the **Generate Report Data** option in the Report and Analysis group to Yes. Previously, a job configured with the option set to No completed without errors, but now, an error is issued.

Multiline update postcode and CASS certification

When your current job has **Disable Certification** in the Non Certified Options group set to No, the **Multiline Update Postcode 2** option in the Standardization Options group must be set to Update or Erase_Then_Update. Previously, a job configured with the option set to Don't Update completed without errors, but now, an error is issued.

An error is also issued under the same situation when the **Multiline Update Postcode 1** option is set to Don't Update.

6.3.20.9 NCOALink Contact Details group

The option list for NCOALink has changed. Previously the Contact Details group did not have any subgroups. In the new release, the Contact Details group is a subgroup under Contact Detail List.

When you upgrade your existing jobs that contain one or more Contact Details groups, a new Contact Detail List option group is added to the transform Options tab. All of your contact details in your existing jobs will transfer to the correct option.

Related Topics

• Reference Guide: Transforms, Data Quality transforms, USA Regulatory Address Cleanse, NCOALink options, Contact detail list

6.3.20.10 NCOALink Use Postcode Order Caching option removed

When you upgrade, the **Use Postcode Order Caching** option in the Transform Performance option group is removed from all existing jobs and from the list of options.

6.3.20.11 Windows Use Extended Memory option removed

The **Windows Use Extended Memory** option in the Transform Performance group is removed. This option was only applicable for 32-bit systems and the new release of SAP BusinessObjects Data Services requires a 64-bit processor.

6.3.20.12 NCOALink memory caching performance

In this release, the amount entered in the **NCOALink Memory Usage** option is the total for all of the threads, not per thread. Therefore, if you want to maintain the performance from previous job setups, you need to adjust this setting appropriately.

Related Topics

• Reference Guide: Transform performance

6.3.20.13 Address_Type OFFICIAL changed to Alias_Type OFFICIAL

When you upgrade existing jobs, the software replaces the output field Address_Type (OFFICIAL) with the new output field Alias_Type (OFFICIAL). The new output field has a length of 1.

6.3.21 GeoCensus and Geocoder

The GeoCensus functionality in the USA Regulatory Address Cleanse transform will be deprecated in a future version. It is recommended that you upgrade any data flows that currently use the GeoCensus functionality to use the Geocoder transform.

To obtain the same results and performance from the Geocoder transform as from the GeoCensus functionality in the USA Regulatory Address Cleanse transform, set the Geocoder transform options and field mappings as directed in the following sections.

For more information on additional features available in the Geocoder transform, see the *Reference Guide* and *Designer Guide*.

Related Topics

- Designer Guide: Data Quality, Geocoding
- Reference Guide: Transforms, Geocoder

6.3.21.1 Directories

The size of the U.S. geocoding directories has significantly increased in this release to support the new reverse and point-of-interest geocoding features. You can purchase and download one of two geocoding directory products for SAP BusinessObjects Data Services XI 4.0:

- SBOP GEO DIR US - NAVTEQ—the `geo_us_nt.dir` file in the `AdvancedGeoFeatures.zip` file contains the range directory, which is approximately 60GB.
- SBOP GEO PARCEL US - NAVTEQ—the `geo_us_nt.dir` file in the `AdvancedGeoFeatures.zip` file contains the range and parcel directory, which is approximately 100GB.

By default, you receive the `geo_us_nt.dir` file, which is the “basic” directory (used if you do not plan to use advanced processing capabilities, such as reverse and point-of-interest geocoding). If you choose to use the range-only directory (approximately 60GB) or the range and parcel directory (approximately 100GB), you must unzip the `AdvancedGeoFeatures.ZIP` file. The `AdvancedGeoFeatures.zip` file contains the large directory (either 60GB or 100GB, depending on the package that you purchase). When the `AdvancedGeoFeatures.zip` is uncompressed, the existing `geo_us_nt.dir` (the default “basic” directory file) is overwritten by one of the two advanced directories. If you would like to switch back to the basic directory in the future, a backup of the basic directory, named `geo_us_nt_basic.dir`, is provided. To use the directory, back up or rename the existing file and rename `geo_us_nt_basic.dir` to `geo_us_nt.dir`.

For the Geocoder transform, all of the U.S. geocoding directories, including the `ageo*.dir` and `cgeo2.dir` files, must be in the same location. If they are currently installed in different locations, you must move them.

For details about the files that are installed with your directory package, see the *Directories* update that is issued with the package.

6.3.21.2 Option settings

Geo Mode option

The following describes the Geocoder transform option values that you must set to replicate the settings for the USA Regulatory Address Cleanse transform's Geo Mode option.

USA Regulatory Address Cleanse Geo Mode option value	Geocoder Best Assignment Level option value	Geocoder Locality Assignment Threshold option value	Geocoder Postcode Assignment Threshold option value
Address	Preferred	None	None
Centroid	Postcode	None	Preferred
Both	Preferred	None	Preferred
None	Postcode	None	None

Enable Geo Only option

The Enable Geo Only option specifies whether the USA Regulatory Address Cleanse transform suppresses address cleansing with the postal directories.

The following describes the data flow configuration that you must use with the Geocoder transform to replicate the settings for the USA Regulatory Address Cleanse transform's Enable Geo Only option.

USA Regulatory Address Cleanse Enable Geo Only option value	Geocoder data flow
Yes	Do not include a USA Regulatory Address Cleanse transform in front of the Geocoder transform in the data flow.
No	Include a USA Regulatory Address Cleanse transform in front of the Geocoder transform in the data flow.

Other options

To obtain the same results and performance in the Geocoder transform as you received with the Data Services USA Regulatory Address Cleanse transform GeoCensus functionality, set the options in the Geocoder transform to the following values. Where the value is n/a, leave the value blank.

Geocoder option	Value
Offset Coordinates	No
Default Max Records	n/a
Default Radius	n/a

6.3.21.3 Input and output fields

To obtain the same results and performance in the Geocoder transform as you received with the Data Services USA Regulatory Address Cleanse transform GeoCensus functionality, map the Geocoder fields as follows.

Map address fields

The Geocoder transform requires discrete fields, such as primary street name, primary premise number, postcode, and so on. The Country field is the only field that is required; however, the more input data you can provide, the better results you will obtain.

If the Enable Geo Only option in the USA Regulatory Address Cleanse transform was set to No, map the discrete USA Regulatory Address Cleanse transform address output fields to the equivalent Geocoder input fields.

If the Enable Geo Only option in the USA Regulatory Address Cleanse transform was set to Yes, the Geocoder transform input fields must be mapped from discrete, parsed input fields from a source other than the USA Regulatory Address Cleanse transform.

Map the Locality field

Map the USA Regulatory Address Cleanse transform's Locality1_Name output field to the Geocoder transform's Locality1 input field.

Add the Country field

The Data Services USA Regulatory Address Cleanse transform does not output the country, because it only processes U.S. data. The Country field must be mapped for the Geocoder transform.

Although your input data may include country data, it may not be populated in all records, and the data content may vary, so the best practice is to use a Query transform to add country data.

To add the Country field where all the records are set to U.S., include a Query transform upstream from the Geocoder transform. Pass all fields through the Query transform and add a new varchar(2) field hardcoded with "US".

Geocoder transform fields

The following fields are supported in the Geocoder transform, and are not available in the Data Services USA Regulatory Address Cleanse transform GeoCensus functionality. If you map them in the Geocoder transform, you will not obtain the same results after you upgrade from the Data Services USA Regulatory Address Cleanse transform to the Geocoder transform.

Input fields

- POI_Name
- POI_Type

Output fields

- Census_Tract_Block_Prev
- Census_Tract_Block_Group_Prev
- Metro_Stat_Area_Code_Prev
- Minor_Div_Code_Prev
- Population_Class_Locality1
- Side_Of_Primary_Address
- Stat_Area_Code_Prev

Related Topics

- Reference Guide: Transforms, Geocoder

6.4 Behavior changes in version 12.2.1

The following sections describe changes in the behavior of SAP BusinessObjects Data Services XI 3.2 Service Pack 1 (12.2.1) from previous releases of Data Services and Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see Data Quality to Data Services Upgrade.

This section includes migration-specific information associated with the following features:

- Microsoft SQL Server 2008 database support
- WSDL version

6.4.1 Microsoft SQL Server 2008 database support

In this release, the Microsoft SQL Server 2008 Native Client ODBC driver is required to support SQL Server 2008 datastores. Existing jobs that use the SQL Server 2008 datastore will not execute successfully without the native driver. After the SQL Server 2008 Native Client ODBC driver is installed, no changes are required to SAP BusinessObjects Data Services datastores or jobs.

6.4.2 WSDL version

The Data Services web services WSDL has been updated to version 2.1. This version adds a Data Services-created target namespace in the published schema for real time services that do not have a target namespace specified. This version also adds an extra attribute to the schema's root element: `elementFormDefault="true"`.

6.5 Behavior changes in version 12.2.0

The following sections describe changes in the behavior of SAP BusinessObjects Data Services XI 3.2 (12.2.0) from previous releases of Data Services and Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see Data Quality to Data Services Upgrade.

This section includes migration-specific information associated with the following features:

- Before upgrading SAP BusinessObjects Data Services or SAP BusinessObjects Enterprise
- Data Cleanse migration considerations
- Updated postal directories
- Updated LACSLink output fields
- New fields added to USA Regulatory Address Cleanse
- "USPS-preferred address"
- Export reports
- Target table editor option tab
- Fixed-width File Format
- Pushdown SQL operations

- Citrix support
- Bundled SAP libraries
- RFC Server support change
- Additional datetime information in overflow file
- Web service security settings
- Unsupported database versions
- UNIX $LINK_DIR directories

6.5.1 Before upgrading SAP BusinessObjects Data Services or SAP BusinessObjects Enterprise

In an environment where SAP BusinessObjects Data Services XI 3.x and SAP BusinessObjects Enterprise XI 3.x co-exist on the same server, ensure that you review the Installation Limitations and Known Issues sections of the *Release Notes* before you uninstall or upgrade either Data Services or BusinessObjects Enterprise.

6.5.2 Data Cleanse migration considerations

During the upgrade, changes were made to the Data Cleanse transform that affect the behavior of existing jobs. Log files detailing the changes are available in the `<LINK_DIR>\Admin\Repo` and `<LINK_DIR>\log` folders.

In addition to the changes detailed in the log file, all dictionaries must be upgraded and additional manual steps may be required to upgrade your existing data flows.

6.5.2.1 Upgrade Data Cleanse dictionaries

Underlying changes to content and structure require that you upgrade all dictionaries. A Data Cleanse job will fail to execute if the dictionary is not of the 3.2 version. For specific upgrade steps, see the Installation Limitations section of the *Release Notes*.

Note:
Before you uninstall your existing version of Data Services, you must export your person_firm dictionary changes or custom Universal Data Cleanse dictionaries.

6.5.2.2 Change in Rule file structure

Beginning with version 3.2, the Data Cleanse engine interacts with the format lines of the referenced rule file to determine the data included in the Person1-6 and Dual_Name1-6 output fields. You may observe differences in the output of these fields.

6.5.2.3 Title information in Person and Dual_Name output fields

Title information is no longer included in the Person and Dual_Name output fields. If you need to include title information in your output, you can add a Query transform to append the title.

6.5.2.4 Deprecated transform options

Certain options were removed from the Data Cleanse transform. Your output for Person and Dual_Name fields will reflect these changes. The following options are no longer available in the transform:

Option Group	Option Name	Required Action
Standardization Options > Person	COMBINE_MAT_HON_POSTNAMES	If you want to create a combined field, add a Query transform after the Data Cleanse transform to append the discrete output fields.
Standardization Options > Person	ASSIGN_MRS	The prename Ms. is now assigned to all strong female names. To modify this term add a Query transform after the Data Cleanse transform and use the search_replace function to replace the terms.

Option Group	Option Name	Required Action
Standardization Options > Person > Greeting Options	NAME CONNECTOR	Remove all "Greeting" output fields.
	NAME_PREFIX	
	NAME_SUFFIX	
	PRENAME_MALE	
	PRENAME_FEMALE	
	NAME_MALE_SALUTATION	
	NAME_FEMALE_SALUTATION	
	NAME_GENERIC_SALUTATION	
	NAME_MULTIPLE_MALE_SALUTATION	
	NAME_MULTIPLE_FEMALE_SALUTATION	
	NAME_MULTIPLE_GENERIC_SALUTATION	
	USE_TITLE_IF_NO_NAME	
	MULTIPLE_STYLE	
	GREETING_TYPE	

6.5.2.5 Deprecated and changed output fields

From each data flow, remove all output fields containing the following GENERATED_FIELD_NAMEs:

- Greeting
- Address
- Last_Line

From each data flow, change the GENERATED_FIELD_CLASS from PARSED to STANDARDIZED for the following GENERATED_FIELD_NAMEs:

- Gender
- Gender_ID
- Rule_Label

- Score
- Email_is_ISP
- All Match_Std fields

From each data flow, change the GENERATED_FIELD_CLASS from STANDARDIZED to PARSED for all output fields with a GENERATED_FIELD_NAME of Extra.

6.5.3 USA Regulatory Address Cleanse transform

6.5.3.1 Updated postal directories

New versions of postal directories used by the USA Regulatory Address Cleanse transform are included in SAP BusinessObjects Data Services XI 3.2.0 (12.2.0). Migration updates existing configurations and jobs to reference the updated directory versions. The updated directories are listed below:

- zip4us.dir
- city10.dir (referenced as city09.dir in previous release)
- zcf10.dir (referenced as zcf09.dir in previous release)
- revzip4.dir
- zip4us.shs
- zip4us.rev

Note:

- You must install the new directory versions before you run the USA Regulatory Address Cleanse transform in SAP BusinessObjects Data Services XI 3.2.0 (12.2.0).
- Do not install or use the directories that are meant for SAP BusinessObjects Data Services XI 3.2.0 (12.2.0) with a previous version of the software. The new directories are not backward compatible.

Related Topics

- Designer Guide: Data quality, Set up the reference files

6.5.3.2 Updated LACSLink output fields

SAP BusinessObjects Data Services introduces a new field class named Pre_LACSLink. With the new field class, the following output fields will be remapped: PRE_LACSLINK_POSTCODE and

PRE_LACSLINK_RECORD_TYPE. Any future projects should include the two existing fields (see table below) to hold the pre-LACSLink information.

Note:
Pre_LACSLink field class type fields retain the address components that were processed prior to the LACSLink processing.

Field	Old field class	Remapped to	New field class
PRE_LACSLINK_POST-CODE	Best	POSTCODE1	Pre-LACSLink
PRE_LACSLINK_POST-CODE	Correct	POSTCODE1	Pre-LACSLink
PRE_LACSLINK_POST-CODE	Move-updated	N/A*	N/A
PRE_LAC-SLINK_RECORD_TYPE	Best	ADDRESS_TYPE	Pre-LACSLink
PRE_LAC-SLINK_RECORD_TYPE	Correct	ADDRESS_TYPE	Pre-LACSLink
PRE_LAC-SLINK_RECORD_TYPE	Move-updated	N/A *	N/A

* The move-updated versions of the PRE_LACSLINK_POSTCODE and PRE_LACSLINK_RECORD_TYPE fields will not migrate to the new version of the software. If you want to delete the field from your current jobs, you must do so manually.

Migration will not change any current jobs that contain the non-supported fields. Instead the software issues a warning that the fields are no longer supported.

Related Topics

- Reference Guide: Data Quality Fields, USA Regulatory Address Cleanse

6.5.3.3 New fields added to USA Regulatory Address Cleanse

New options have been added to the PAF Details, Contact Details, and USPS License Information groups of fields for the USA Regulatory Address Cleanse transform.

Option	Option group	Description
Email of person signing	PAF Details	The email address for the person who is signing the PAF. This parameter is optional.
Company web-site	PAF Details	The company website address for the person signing the PAF. This parameter is optional.
Using Alterna-tive PAF	PAF Details	Select Yes if you are using a PAF that is not the USPS form (you must have permission from the USPS).
Contact compa-ny website	Contact Details	The website of the broker or list administrator. This parameter is optional.
IMB Mailer ID	USPS License Information	Holds your Intelligent Mail Barcode number, if applicable.

Related Topics

- Reference Guide: USA Regulatory Address Cleanse

6.5.3.4 Preferred Address

Migration sets **Address Line Alias** to Convert when your existing USA Regulatory Address Cleanse jobs are in CASS mode (**Disable Certification** is set to No). Migration will not update your existing USPS Regulatory Address Cleanse jobs when **Disable Certification** or **Address Line Alias** options use a substitution variable. Instead you will receive a warning message.

Related Topics

- Reference Guide: Standardization options

6.5.4 Export reports

Export reports

Previously, to view and save data quality reports, you had to view each report in the Management Console and export it individually. The new Export Data Quality Reports option lets you generate and

export all of the specified job reports at once upon execution. You configure the report server by using the new Report Server Configuration node in the Management Console Administrator.

If you have an existing repository from a release prior to SAP BusinessObjects Data Services XI 3.2 and you want to export reports, you must save the report server configuration for the repository. To save the report server configuration:

1. From the Management Console, select **Administrator > Web Services**, and click the **Web Services Configuration** tab.
2. Ensure the **Export_DQReport** session security is disabled.
3. To save the report server configuration, do one of the following:
   - Select **Management > Report Server Configuration**, select the repository, and click **Apply**.

     This method lets you change the default export parameters, such as the location.
   - Select **Management > Repositories**, select the repository, and click **Apply**.

     This method uses the default export parameters.

6.5.5 Target table editor option tab

The changes made to the Options tab in the editor for database target tables have rearranged previously available options. The options are arranged in a logical order and the settings are easier to view. During migration, the options in your current jobs will transfer to the corresponding options in the re-designed Options tab.

6.5.6 Fixed-width File Format

This release of SAP BusinessObjects Data Services 12.2 supports both byte and character data alignment for fixed-width file formats. If you want to change your alignment from character to bytes, set the alignment in the new **Data Alignment** option (in File Format, General options group).

- Data Services versions 12.0 and 12.1 support character data alignment only.
- Data Integrator versions 11.7 or older support byte data alignment only.

When you migrate from 12.0 to the new version 12.2, migration changes your existing jobs based on the criteria below:

- If you are migrating your existing fixed-width files from Data Services 12.0 or later, migration sets the new option to Character.
- If you are migrating your existing fixed-width files from Data Services 11.7 or older, migration sets the new option to Byte.

Related Topics

- Reference Guide, File Format

6.5.7 Pushdown SQL operations

Beginning with the SAP BusinessObjects Data Services XI 3.2 release, the Optimizer can include tables with different schema names in push-down operations. The tables must be from datastores that have the same database type and the same connection name, server name, or data source name.

In existing jobs, a data access permission error may occur if the user does not have access privileges for all datastores included in the push-down operation. If the permission error occurs, do one of the following:

- Open the data flow "Properties" window and deselect the **Use database links** checkbox.
- As appropriate, request access privileges from your Administrator.

6.5.8 Citrix support

In previous SAP BusinessObjects Data Services releases, when the Designer or Locale Selector is started, many libraries must be copied from the installation directory to a user-specific resource directory. With this release, only user-specific configuration files are copied and stored in a user-specific resource directory. All required runtime libraries are referenced from the main installation directory.

Previously, a sample DOS script template was provided and manually edited to publish the Designer.

When migrating, existing published Designer and Locale Selector applications need to be reconfigured and republished in Citrix. You can use two newly provided DOS scripts to reconfigure and republish the applications. The scripts are included as part of the SAP BusinessObjects Data Services client component installation.

By default, the scripts create the user-specific resource directories in the C:\Users directory. You can override the default location by specifying a different path as a parameter on the command line in Citrix.

The new DOS scripts are:

- Designer: RunDSDesignerMU.bat `<user's resource location>`
- Locale Selector: RunLocaleSelectorMU.bat `<user's resource location>`

For more information, see the Citrix Support section of the *Installation Guide for Windows*.

6.5.9 Bundled SAP libraries

Data Services now automatically installs the `librfc.dll` library required for SAP connectivity. Its default installation location is in `<LINK_DIR>\bin` and it is used by the Data Services engine, Designer, and Access Server. Separate installation of the library is no longer required.

The SAP Java Connector library is also automatically installed for all supported platforms. Its default installation location is in a platform-dependent subdirectory in `<LINK_DIR>/bin/jco3`, and it is used by the new RFC Server Interface in the SAP Connections node of the Administrator.

6.5.10 RFC Server support change

Beginning with SAP BusinessObjects Data Services XI 3.2, it is recommended that you use the new RFC Server Interface in the Administrator.

6.5.11 Additional datetime information in overflow file

In Microsoft SQL Server target tables, when the **Use overflow file** option is enabled and rows that fail to load are written to a file, subsecond information is now included for data of the datetime and time data types. The exact precision is defined in the database and the subsecond information is included, even when the value is 0. For example, in previous versions of the software, an entry in the overflow file might have been 2008-01-03 12:12:12; beginning with version 3.2, the same entry will be 2008-01-03 12:12:12.000.

6.5.12 Web service security settings

To ensure the security of your software, the security setting is enabled by default for all web service operations. Be aware that this may cause your existing web service calls to fail. To verify your security settings, log into the Management Console, select **Administrator > Web Services**, and click the **Web Services Configuration** tab.

6.5.13 Unsupported database versions

As of version 12.2.0, SAP BusinessObjects Data Services no longer allows you to select a datastore configuration or source that refers to one of the following database versions:

- IBM DB2 UDB 6.1
- IBM DB2 UDB 7.1
- IBM DB2 UDB 7.2
- Microsoft SQL Server 7.0
- Netezza NPS 2.5
- Oracle 8.0
- Oracle 8.1
- Sybase ASE 11.x
- Sybase IQ 12.5
- Teradata 2.5

You must migrate to a supported database version by creating new datastore configurations based on the obsolete configurations. For supported database versions, , see the *Product Availability Matrix* document located in the SAP Service Marketplace: https://service.sap.com/PAM.

6.5.14 UNIX $LINK_DIR directories

For UNIX systems, after migrating from Data Integrator 11.x to SAP BusinessObjects Data Services 12.2.0, the Data Integrator and Data Services versions of the `$LINK_DIR` directory co-exist. The new directory is named `dataservices`.

If you had configured an Access Server (including real-time services, if any) using a folder in the Data Integrator 11.x `$LINK_DIR` directory, that Access Server continues to access the configuration files from that path after you migrate to SAP BusinessObjects Data Services 12.2.0. If you configure a new Access Server or edit the existing one to use a new folder, then you won't need the previous Access Server folder.

In the Management Console, views of job execution history that occurred before migration are based on data in the Data Integrator`$LINK_DIR/log` directory. The views of job execution history that occurred after migration are based on data in the Data Services `$LINK_DIR/log` directory.

If you choose to delete the Data Integrator`$LINK_DIR` directory, you will lose the execution logs for job execution that occurred before migration as well as the Access Server configuration files configured in the Data Integrator`$LINK_DIR` directory.

6.6 Behavior changes in version 12.1.1

The following sections describe changes in the behavior of Data Services 12.1.1 from previous releases of Data Services and Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see Data Quality to Data Services Upgrade.

This section includes migration-specific information associated with the following features:

- Netezza table name syntax change
- Netezza bulk loading
- Blob data type enhancements
- Neoview bulk loading

6.6.1 Netezza table name syntax change

For Netezza datastores, the table name used in SQL statements generated by Data Services is qualified using the following syntax: *database_name..table_name* (note the double dot between the two names). Prior to Data Services 12.1, the syntax was *owner_name.table_name*. The syntax change was made to enable Data Services to push down the SQL statement across multiple databases.

For Netezza datastores created in previous versions of the software, tables that were imported into Data Services using the owner name instead of the database name are qualified using the older syntax. The SQL statement that Data Services generates will be incorrect. To resolve this issue, do one of the following:

- If the table owner name refers to a datastore alias, change the alias value to the database name.
- If the owner name of the table refers to the database user name, use the **Rename owner** command to change the owner name to the database name or create an alias in the Netezza datastore and set the value of the owner name to the database name.

6.6.2 Netezza bulk loading

In Netezza bulk load settings, the null indicator value has changed from an empty string ("") to "NULL". This change allows empty strings ("") to load in the target table as empty strings rather than null.

In VARCHAR columns where the data itself is the string "NULL", the data will be loaded as null in the target table. In order to load the data "NULL" as a string "Null", set the escape character value to backslash (\) in the Netezza bulk loader option.

Note:
The null indicator string is not case-sensitive.

6.6.3 Neoview bulk loading

If you plan to bulk load data to a Neoview database, we recommend that you set **Timeout** to 1000 in your Neoview target table.

- If you create a new repository in version 12.1.1, you do not need to set **Timeout** because its default value is 1000.
- If you use a 12.1.0 repository when you install version 12.1.1, the default value for **Timeout** is 60. Therefore, increase **Timeout** to 1000 for new data flows that bulk load into a Neoview database.

Related Topics

- Reference Guide: Objects, HP Neoview target table options

6.6.4 Blob data type enhancements

Data Services 12.1.1 provides the following enhancements for binary large object (blob) data types:

- You can now define blob data type columns in a fixed-width file format, and you can read from and load to blob columns in fixed-width files
- The `dqmigration` utility now migrates Data Quality binary data types in fixed-width flat files to Data Services blob (instead of varchar) data types in fixed-width file formats. You no longer need to change the data type from varchar to blob after migration.

In a fixed-width file, the blob data is always inline with the rest of the data in the file. The term “inline” means the data itself appears at the location where a specific column is expected.

The 12.1.0 release of Data Services introduced support for blob data types in a delimited file. In a delimited file, the blob data always references an external file at the location where the column is expected. Data Services automatically generates the file name.

The following table summarizes the capabilities that each release provides for blob data types:

File Type	Inline		«Filename»	
	12.1.0	12.1.1	12.1.0	12.1.1
blob in delimited file	No	No	Yes	Yes
blob in fixed-width file	No	Yes	No	No

These capabilities help customers migrate their existing Data Quality projects that handle binary data in flat files to Data Services fixed-width file formats. The Data Services blob data type now supports blob data types from Data Quality XI R2 and legacy Firstlogic products.

Related Topics

- Reference Guide: Data Types, blob

6.7 Behavior changes in version 12.1.0

The following sections describe changes in the behavior of Data Services 12.1.0 from previous releases of Data Services and Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see the *Data Quality to Data Services Migration Guide*.

This section includes migration-specific information associated with the following features:

- Cleansing package changes
- DTD-to-XSD conversion
- Minimum requirements for international addressing directories
- Try/catch exception groups
- Upgrading from version 12.0.0 to version 12.1.0

6.7.1 Cleansing package changes

Global Parsing Options have been renamed cleansing packages.

You can no longer use the Global Parsing Options installer to install cleansing packages directly into the repository. You must now use a combination of the cleansing package installer and the Repository Manager instead.

If you have made any changes to your existing cleansing package dictionaries, you must do the following:

1. Export the changes using **Export Dictionary Changes** in the **Dictionary** menu of the Data Services Designer.
2. Install the latest cleansing package.
3. Use the Repository Manager to load the cleansing package into the data cleanse repository.
4. Import the changes into the new cleansing package using **Bulk Load** in the **Dictionary** menu in the Designer.

Related Topics

- Designer Guide: Data Quality, To export dictionary changes
- Designer Guide: Data Quality, To import dictionary changes

6.7.2 DTD-to-XSD conversion

Data Services no longer supports publishing a DTD-based real-time job as a Web service if the job uses a DTD to define the input and output messages.

If you migrate from Data Services 12.0.0 to version 12.1.0, you do not need to do anything unless you change the DTD. If you change the DTD, reimport it to the repository and publish the Web service as in the following procedure.

If you migrate from Data Integrator 11.7 or earlier versions to Data Services 12.1.0 and publish a DTD-based real-time job as a Web service, you must reimport the Web service adapter function because the Web address changed for the Management Console in version 12.0.0. Therefore, you must do the following after you upgrade your repository to version 12.1.0:

1. Use any DTD-to-XSD conversion tool to convert the DTD to XSD.
2. Use the Designer to import the XSD to the Data Services repository.
3. Open the original data flow that is using the DTD and replace it with XSD.
4. Publish the real-time job as Web service.
5. Reimport the service as a function in the Web Service datastore.

Related Topics

- Data Services web address

6.7.3 Minimum requirements for international addressing directories

Due to additional country support and modified database structures (for performance tuning), the minimum disk space requirement for the international addressing directories (All World) has increased as follows:

- For the Global Address Cleanse transform (`ga_country.dir`, `ga_loc12_gen.dir`, `ga_loc12_gen_nogit.dir`, `ga_loc34_gen.dir`, `ga_region_gen.dir`), the minimum requirement has increased from 647 MB to 2.71 GB.
- If you purchase all countries, the disk space requirement has increased from 6.1 GB to 9.34 GB.

6.7.4 Try/catch exception groups

This version of Data Services provides better defined exception groups of errors, new exception groups, and an enhanced catch editor that allows you to select multiple exception groups in one catch to consolidate actions.

After you upgrade your repository to version 12.1, your try/catch blocks created in prior versions contain the 12.1 exception group names and numbers. Be aware of the following situations and additional actions that you might need to take after you upgrade to version 12.1:

- The repository upgrade will map Parser errors (1) and Resolve errors (2) to Execution errors (1000) and will map email errors(16) to System Resource errors (1008). You need to re-evaluate all the actions that are already defined in all the catch blocks and modify them as appropriate, based on the new catch exception group definitions. See the tables below for the mapping of exception groups from version 12.0 to version 12.1 and for the definitions of new exception groups.
- All recoverable jobs in a pre-12.1 system lose their recoverable state when you upgrade. After you upgrade to version 12.1, you need to run the job from the beginning.
- If you upgrade a central repository, only the latest version of a work flow, data flow audit script, and user function contain the 12.1 exception group names. Older versions of these objects contain the pre-12.1 exception group names.
- In version 12.1, if you have a sequence of catch blocks in a workflow and one catch block catches an exception, the subsequent catch blocks will not be executed. For example, if your work flow has the following sequence and Catch1 catches an exception, then Catch2 and CatchAll will not execute. In prior versions, both Catch1 and CatchAll will execute.

```
Try > DataFlow1 > Catch1 > Catch2 > CatchAll
```

Note:
If you import pre-12.1 ATL files, any catch objects will not contain the new exception group names and numbers. Only a repository upgrade converts the pre-12.1 exception groups to the 12.1 exception group names and numbers.

The following table shows how the exception groups in version 12.0 map to the exception groups in version 12.1:

12.0 Exception group (group number)	12.0 Description	12.1 Exception group (group number)	12.1 Description
Catch All Exceptions	All errors	All exceptions	All errors
Parser Errors (1)	Errors encountered while parsing the language	Pre-execution errors (1000)	Parser errors are not caught because parsing occurs prior to execution.
Resolver Errors (2)	Errors encountered while validating the semantics of Data Services objects which have recommended resolutions	Pre-execution errors (1000)	Resolver errors are not caught because parsing occurs prior to execution.
Execution Errors (5)	Internal errors that occur during the execution of a data movement specification	Execution errors (1001)	Errors from the Data Services job server or transforms
Database Access Errors (7)	Generic Database Access Errors	Database Access Errors (1002)	Errors from the database server while reading data, writing data, or bulk loading to tables
File Access Errors (8)	Errors accessing files through file formats	Flat file processing errors (1004)	Errors processing flat files
		File Access Errors (1005)	Errors accessing local and FTP files
Repository Access Errors (10)	Errors accessing the Data Services repository	Repository access errors (1006)	Errors accessing the Data Services repository
Connection and bulk loader errors (12)	Errors connecting to database servers and bulk loading to tables on them	Database Connection errors (1003)	Errors connecting to database servers
Predefined Transforms Errors (13)	Predefined transform errors	R/3 system errors (1007)	Errors while generating ABAP programs, during ABAP generated user transforms, or while accessing R/3 system using its API
ABAP Generation Errors (14)	ABAP generation errors		
R/3 Execution Errors (15)	R/3 execution errors		
Email Errors (16)	Email errors	System Resource errors (1008)	Errors while accessing or using operating system resources, or while sending emails
System Exception Errors (17)	System exception errors		
Engine Abort Errors (20)	Engine abort errors	Execution errors (1001)	Errors from the Data Services job server or transforms

The following table shows the new exception groups in version 12.1:

New 12.1 Exception group (group number)	Description
SAP BW execution errors (1009)	Errors from the SAP BW system.
XML processing errors (1010)	Errors processing XML files and messages
COBOL copybook errors (1011)	Errors processing COBOL copybook files
Excel book errors (1012)	Errors processing Excel books
Data Quality transform errors (1013)	Errors processing Data Quality transforms

6.7.5 Upgrading from version 12.0.0 to version 12.1.0

If you are installing version 12.1.0 and the installer detects a previous installation of version 12.0, you will be prompted to first uninstall version 12.0. The installer will maintain your configuration settings if you install in the same directory.

If you are installing version 12.1.0 on top of version 11.x, you do not need to uninstall the previous version.

6.8 Behavior changes in version 12.0.0

The following sections describe changes in the behavior of Data Services 12.0.0 from previous releases of Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

If you are migrating from Data Quality to Data Services, see the *Data Quality to Data Services Migration Guide*.

This section includes migration-specific information associated with the following features:

- Case transform enhancement
- Data Quality projects in Data Integrator jobs
- Data Services web address
- Large object data type enhancements
- License keycodes
- Locale selection
- ODBC bigint data type

- Persistent and pageable cache enhancements
- Row delimiter for flat files

6.8.1 Case transform enhancement

In this version, you can choose the order of Case expression processing to improve performance by processing the less CPU-intensive expressions first. When the **Preserve expression order** option is not selected in the Case transform, Data Services determines the order to process the case expressions. The **Preserve expression order** option is available only when the **Row can be TRUE for one case only** option is selected.

By default, the **Row can be TRUE for one case only** option is selected and the **Preserve expression order** option is not selected. Therefore, when you migrate to this version, Data Services will choose the order to process your Case expressions by default.

However, the reordering of expressions can change your results because there is no way to guarantee which expression will evaluate to TRUE first. If your results changed in this version and you want to obtain the same results as prior versions, select the **Preserve expression order** option.

6.8.2 Data Quality projects in Data Integrator jobs

To do data cleansing in version Data Integrator 11.7, you created a Data Quality datastore and imported integrated batch projects as Data Quality transforms. When these imported Data Quality transforms were used in an 11.7 job, the data was passed to Data Quality for cleansing, and then passed back to the Data Integrator job.

In Data Services 12, the Data Quality transforms are built in. Therefore, if you used imported Data Quality transforms in Data Integrator 11.7, you must replace them in Data Services with the new built-in Data Quality transforms.

Related Topics

- Modifying Data Integrator 11.7 Data Quality projects
- Migrating Data Quality integrated batch projects
- How integrated batch projects migrate

6.8.3 Data Services web address

In this release, Data Integrator has become part of Data Services. Therefore, the Web address has changed for the Management Console. In previous releases, the Web address used "diAdmin" as the following format shows:

```
http://computername:port/diAdmin
```

In Data Services, the Web address uses DataServices:

```
http://computername:port/DataServices
```

Therefore, when you migrate to Data Services you must make changes in the following situations:

- If you created a bookmark that points to the Management Console in a previous release, you must update the bookmark to the changed Web address.
- If you generated a Web Service Definition Language (WSDL) file in a previous version of Data Integrator, you must regenerate it to use the changed Web address of the Administrator.

6.8.4 Large object data type enhancements

Data Services 12.0 extends the support of large objects as follows:

- Adds support for binary large object (blob) data types from the currently supported database systems (Oracle, DB2, Microsoft SQL Server, and so on).
- Extends support for character large object (clob) and national character object (nclob) data types to other databases.

Prior versions treat the clob and nclob data types as long data types, and this version continues to treat them as long data types.

The following table shows the large data types that version 11.7 supports as long data types and the additional large data types that version 12 now supports. If your pre-version 12 jobs have sources that contain these previously unsupported large data types and you now want to use them in version 12, you must re-import the source tables and modify your existing jobs to select these newly supported data types.

Table 6-16: Database large object data types supported

Database	Database data type	Category	Version 11.7 supports	Version 12.0 supports	Version 12.0 data type
DB2	LONG VARCHAR	CLOB	Yes	Yes	LONG
	CLOB	CLOB	Yes	Yes	LONG
	LONG VARGRAPHIC	NCLOB	No	Yes	LONG
	DBCLOB	NCLOB	No	Yes	LONG
	BLOB	BLOB	No	Yes	BLOB
Informix	LVARCHAR	VARCHAR	Yes	Yes	VARCHAR
	TEXT	CLOB	Yes	Yes	LONG
	BYTE	BLOB	No	Yes	BLOB
	CLOB	CLOB	Yes	Yes	LONG
	BLOB	BLOB	No	Yes	BLOB
Microsoft SQL Server	TEXT	CLOB	Yes	Yes	LONG
	NTEXT	NCLOB	No	Yes	LONG
	VARCHAR (max)	CLOB	No	Yes	LONG
	NVARCHAR (max)	NCLOB	No	Yes	LONG
	IMAGE	BLOB	No	Yes	BLOB
	VARBINARY(max)	BLOB	No	Yes	BLOB
MySQL	TEXT	CLOB	Yes	Yes	LONG
	BLOB	BLOB	No	Yes	BLOB
ODBC	SQL_LONGVARCHAR	CLOB	Yes	Yes	LONG
	SQL_WLONGVARCHAR	NCLOB	No	Yes	LONG
	SQL_LONGVARBINARY	BLOB	No	Yes	BLOB

Database	Database data type	Category	Version 11.7 supports	Version 12.0 supports	Version 12.0 data type
Oracle	LONG	CLOB	Yes	Yes	LONG
	LONGRAW	BLOB	No	Yes	BLOB
	CLOB	CLOB	Yes	Yes	LONG
	NCLOB	NCLOB	Yes	Yes	LONG
	BLOB	BLOB	No	Yes	BLOB
Sybase ASE	TEXT	CLOB	No	Yes	LONG
	IMAGE	BLOB	No	Yes	BLOB
Sybase IQ 12.6 or later	LONG VARCHAR	CLOB	Yes	Yes	LONG
	LONG BINARY	BLOB	No	Yes	BLOB
Teradata	LONG VARCHAR	CLOB	Yes	Yes	LONG
	CLOB	CLOB	Yes	Yes	LONG
	BLOB	BLOB	No	Yes	BLOB

6.8.5 License keycodes

In this version, Data Services incorporates the BusinessObjects Enterprise installation technology and uses keycodes to manage the licenses for the different features. Therefore, Data Services does not use .lic license files anymore but manages keycodes in the License Manager.

6.8.6 Locale selection

In this version, you no longer set the locale of the Job Server when you install Data Services. After installation, the locale of the Job Server is set to **<default>** which enables Data Services to automatically set the locale for the repository connection (for the Designer) and to process job data (for the Job Server) according to the locale of the datastore or operating system. This capability enables Data Services to automatically change the locale for better performance (for example, set the locale to non-UTF-8 if the datastore is non-Unicode data).

The following table shows different datastores and Job Server locale settings, the resulting locale that prior versions set, and the new locale that version 12.0 sets for the data flow. In this table, the Job Server locale is set to **<default>** and derives its value from the operating system.

Datastore 1 locale	Datastore 2 locale	Job Server locale	Data flow locale in prior version	Data flow locale in version 12.0
Single-byte code page	Multi-byte code page	Single-byte code page or Multi-byte code page	Same locale as Job Server	Unicode
Multi-byte code page	Multi-byte code page	Single-byte code page	Single-byte code page	Unicode
Multi-byte code page	Multi-byte code page	Multi-byte code page	Unicode	Unicode
Single-byte code page 1	Single-byte code page 2	Single-byte code page 3	Single-byte code page 3	Unicode
Single-byte code page 1	Single-byte code page 2	Multi-byte code page	Unicode	Unicode
Single-byte code page 3	Single-byte code page 3	Single-byte code page 1	Single-byte code page 1	Single-byte code page 3
Single-byte code page 3	Single-byte code page 3	Multi-byte code page	Unicode	Unicode

The following table summarizes the locale that Data Services now sets for each data flow when the locale of the Job Server is set to **<default>**. Different data flows in the same job can run in either single-byte or Unicode.

Locale of datastores in data flow	Job Server locale	Locale that Data Services sets
One datastore has multi-byte locale	Single-byte or Multi-byte	Unicode
Different single-byte locales	Single-byte or Multi-byte	Unicode
Same single-byte locale	Single-byte	Single-byte
Same single-byte locale	Multi-byte	Unicode

You can override the default locale for the Job Server by using the Data Services Locale Selector utility. From the Windows **Start** menu, select **Programs** > **BusinessObjects XI 3.2** > **BusinessObjects Data Services** > **Data Services Locale Selector**.

6.8.7 ODBC bigint data type

For an ODBC datastore, Data Services now imports a bigint data type as decimal. In prior releases of Data Integrator, the bigint data type was imported as a double data type. If your pre-version 12 jobs have sources that contain bigint data types, you must re-import the source tables and modify your existing jobs to handle them as decimal data types.

6.8.8 Persistent and pageable cache enhancements

This release of Data Services provides performance enhancements for the persistent and pageable caches. Decimal data types now use only half the memory used in prior versions.

However, persistent cache tables created in prior versions are not compatible with Data Services. You must recreate them by rerunning the jobs that originally created and loaded the target persistent cache tables.

6.8.9 Row delimiter for flat files

In Data Services 12, you can now specify the following values as row delimiters for flat files:

- {new line}

  If you specify this value for the row delimiter, Data Services writes the appropriate characters for the operating system on which the Job Server is running:
  - CRLF (\r\n) in Windows
  - LF (\n) in UNIX
- any character sequence

  In this case, Data Services writes the characters you entered.
- {UNIX new line}

  In this case, Data Services writes the characters LF (\n) regardless of the operating system.
- {Windows new line}

  In this case, Data Services writes the characters CRLF (\r\n) regardless of the operating system.

In previous releases, you could only specify the following values as row delimiters for flat files, and the behavior is the same as in the new release:

- {new line}
- any character sequence

If your target appends to an existing file that was generated in a prior release, Data Services is not backward compatible for the following situations:

- Your Job Server runs on a Windows platform and you choose {UNIX new line} for the row delimiter.
- Your Job Server runs on a UNIX system and you choose {Windows new line} for the row delimiter.

In these situations, you must define a new file format, load data from the existing file into the new file specifying the new row delimiter, and then append new data to the new file with the new row delimiter.

6.9 Behavior changes in version 11.7.3

The following sections describe changes in the behavior of Data Services 12.0 from previous releases of Data Integrator. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

This section includes migration-specific information associated with the following features:

- Data flow cache type
- Job Server enhancement
- Logs in the Designer
- Pageable cache for memory-intensive data flows

6.9.1 Data flow cache type

When upgrading your repository from versions earlier than 11.7 to an 11.7 repository using version 11.7.3.0, all of the data flows will have a default **Cache type** value of `pageable`. This is different from the behavior in 11.7.2.0, where the upgraded data flows have a default **Cache type** value of `in-memory`.

6.9.2 Job Server enhancement

Using multithreaded processing for incoming requests, each Data Integrator Job Server can now accommodate up to 50 Designer clients simultaneously with no compromise in response time. (To accommodate more than 50 Designers at a time, create more Job Servers.)

In addition, the Job Server now generates a Job Server log file for each day. You can retain the Job Server logs for a fixed number of days using a new setting on the Administrator Log retention period page.

6.9.3 Logs in the Designer

In Data Integrator 11.7.3, you will only see the logs (trace, error, monitor) for jobs that started from the Designer, not for jobs started via other methods (command line, real-time, scheduled jobs, or Web services). To access these other log files, use the Administrator in the Data Integrator Management Console.

6.9.4 Pageable cache for memory-intensive data flows

As a result of multibyte metadata support, Data Integrator might consume more memory when processing and running jobs. If the memory consumption of some of your jobs were running near the 2-gigabyte virtual memory limit in a prior version, there is a chance that the same jobs could run out of virtual memory. If your jobs run out of memory, take the following actions:

- Set the data flow **Cache type** value to `pageable`.
- Specify a pageable cache directory that:
  - Contains enough disk space for your data. To estimate the amount of space required for pageable cache, consider factors such as the number of concurrently running jobs or data flows and the amount of pageable cache required for each concurrent data flow
  - Exists on a separate disk or file system from the Data Integrator system and operating system (such as the C: drive on Windows or the root file system on UNIX).

6.9.5 Adapter SDK

The Adapter SDK no longer supports native SQL or partial SQL.

6.9.6 PeopleSoft 8

PeopleSoft 8 support is implemented for Oracle only.

Data Integrator jobs that ran against previous versions of PeopleSoft are not guaranteed to work with PeopleSoft 8. You must update the jobs to reflect metadata or schema differences between PeopleSoft 8 and previous versions.

6.10 Behavior changes in version 11.7.2

The following sections describe changes in the behavior of Data Integrator 11.7.2 from previous releases. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

This section includes migration-specific information associated with the following features:

- Embedded data flows
- Oracle Repository upgrade
- Solaris and AIX platforms

6.10.1 Embedded data flows

In this version of Data Integrator, you cannot create embedded data flows which have both an input port and an output port. You can create a new embedded data flow only at the beginning or at the end of a data flow with at most one port, which can be either an input or an output port.

However, after upgrading to Data Integrator version 11.7.2, embedded data flows created in previous versions will continue to run.

6.10.2 Oracle Repository upgrade

If you previously upgraded your repository to Data Integrator 11.7.0 and open the "Object State Report" on the Central repository from the Web Administrator, you may see the error "ORA04063 view ALVW_OBJ_CINOUT has errors". This occurs if you had a pre-11.7.0. Oracle central repository and upgraded the central repository to 11.7.0.

Note:
If you upgraded from a pre-11.7.0.0 version of Data Integrator to version 11.7.0.0 and you are now upgrading to version 11.7.2.0, this issue may occur, and you must follow the instructions below. Alternatively, if you upgraded from a pre-11.7.0.0 version of Data Integrator to 11.7.2.0 without upgrading to version 11.7.0.0, this issue will not occur and has been fixed in 11.7.2.0.

To fix this error, manually drop and recreate the view `ALVW_OBJ_CINOUT` using an Oracle SQL editor, such as SQLPlus.

Use the following SQL statements to perform the upgrade:

```
DROP VIEW ALVW_OBJ_CINOUT;

CREATE VIEW ALVW_OBJ_CINOUT (OBJECT_TYPE, NAME, TYPE, NORMNAME, VERSION,
DATASTORE, OWNER,STATE, CHECKOUT_DT, CHECKOUT_REPO, CHECKIN_DT,

CHECKIN_REPO, LABEL, LABEL_DT,COMMENTS,SEC_USER,SEC_USER_COUT) AS

(

select OBJECT_TYPE*1000+TYPE,NAME, N'AL_LANG' , NORMNAME,VERSION,DATASTORE,
OWNER, STATE, CHECKOUT_DT, CHECKOUT_REPO, CHECKIN_DT,

CHECKIN_REPO, LABEL, LABEL_DT,COMMENTS,SEC_USER ,SEC_USER_COUT

from AL_LANG L1 where NORMNAME NOT IN ( N'CD_DS_D0CAFAE2' , N'XML_TEMPLATE_FOR
MAT' , N'CD_JOB_D0CAFAE2' , N'CD_DF_D0CAFAE2' , N'DI_JOB_AL_MACH_INFO' ,
N'DI_DF_AL_MACH_INFO' , N'DI_FF_AL_MACH_INFO' )

union

select 20001, NAME,FUNC_TYPE ,NORMNAME, VERSION, DATASTORE, OWNER, STATE,
CHECKOUT_DT, CHECKOUT_REPO, CHECKIN_DT,

CHECKIN_REPO, LABEL, LABEL_DT,COMMENTS,SEC_USER ,SEC_USER_COUT

from AL_FUNCINFO F1 where FUNC_TYPE = N'User_Script_Function' OR OWNER <>
N'acta_owner'

union

select 30001, NAME, N'PROJECT' , NORMNAME, VERSION, N'' , N'' , STATE,
CHECKOUT_DT, CHECKOUT_REPO, CHECKIN_DT,

CHECKIN_REPO, LABEL, LABEL_DT,COMMENTS,SEC_USER ,SEC_USER_COUT

from AL_PROJECTS P1

union

select 40001, NAME,TABLE_TYPE, NORMNAME, VERSION, DATASTORE, OWNER, STATE,
CHECKOUT_DT, CHECKOUT_REPO, CHECKIN_DT,

CHECKIN_REPO, LABEL, LABEL_DT,COMMENTS,SEC_USER ,SEC_USER_COUT

from AL_SCHEMA DS1 where DATASTORE <> N'CD_DS_d0cafae2'

union

select 50001, NAME, N'DOMAIN' , NORMNAME, VERSION, DATASTORE, N'' , STATE,
CHECKOUT_DT, CHECKOUT_REPO, CHECKIN_DT,
```

```
CHECKIN_REPO, N'' ,to_date( N'01/01/1970' , N'MM/DD/YYYY' ), N'' ,SEC_USER
,SEC_USER_COUT

from AL_DOMAIN_INFO D1

);
```

6.10.3 Solaris and AIX platforms

Data Integrator 11.7.2 on Solaris and AIX platforms is a 64-bit application and requires 64-bit versions of the middleware client software (such as Oracle and SAP) for effective connectivity. If you are upgrading to Data Integrator 11.7.2 from a previous version, you must also upgrade all associated middleware client software to the 64-bit version of that client. You must also update all library paths to ensure that Data Integrator uses the correct 64-bit library paths.

6.11 Behavior changes in version 11.7.0

The following sections describe changes in the behavior of Data Integrator 11.7.0 from previous releases. In most cases, the new version avoids changes that would cause existing applications to modify their results. However, under some circumstances a change has been deemed worthwhile or unavoidable.

This section includes migration-specific information associated with the following features:

- Data Quality
- Distributed data flows
- JMS Adapter interface
- XML Schema enhancement
- Password management
- Repository size
- Web applications
- Web services

6.11.1 Data Quality

Data Integrator 11.7.0 integrates the BusinessObjects Data Quality XI application for your data quality (formerly known as Data Cleansing) needs, which replaces Firstlogic's RAPID technology.

Note:
The following changes are obsolete with Data Services version 12.0 because the Data Quality transforms are built into Data Services, and you can use them just like the regular Data Integrator transforms in a data flow.

The following changes to data cleansing occurred in Data Integrator 11.7.0:

- Depending on the Firstlogic products you owned, you previously had up to three separate transforms that represented data quality functionality: Address_Enhancement, Match_Merge, and Name_Parsing.

  Now, the data quality process takes place through a Data Quality Project. To upgrade existing data cleansing data flows in Data Integrator, replace each of the cleansing transforms with an imported Data Quality Project using the Designer.

  You must identify all of the data flows that contain any data cleansing transforms and replace them with a new Data Quality Project that connects to a Data Quality blueprint or custom project.
- Data Quality includes many example blueprints - sample projects that are ready to run or can serve as a starting point when creating your own customized projects. If the existing blueprints do not completely suit your needs, just save any blueprint as a project and edit it. You can also create a project from scratch.
- You must use the Project Architect (Data Quality's graphical user interface) to edit projects or create new ones. Business Objects strongly recommends that you do not attempt to manually edit the XML of a project or blueprint.
- Each imported Data Quality project in Data Integrator represents a reference to a project or blueprint on the data quality server. The Data Integrator Data Quality projects allow field mapping.

6.11.1.1 To migrate your data flow to use the new Data Quality transforms

Note:
The following procedure is now obsolete with Data Services version 12.0 because the Data Quality transforms are now built into Data Services and you can use them just like the regular Data Integrator transforms in a data flow. If you performed this procedure in Data Integrator version 11.7, you will need to migrate these data flows to Data Services. See Data Quality projects in Data Integrator jobs .

1. Install Data Quality XI, configure and start the server. For installation instructions, see your Data Quality XI documentation.

   Note:
   You must start the server before using Data Quality XI with Data Integrator.

2. In the Data Integrator Designer, create a new Business Objects Data Quality datastore and connect to your Data Quality server.
3. Import the Data Quality projects that represent the data quality transformations you want to use. Each project appears as a Data Quality project in your datastore. For the most common data quality transformations, you can use existing blueprints (sample projects) in the Data Quality repository

4. Replace each occurrence of the old data cleansing transforms in your data flows with one of the imported Data Quality transforms. Reconnect the input and output schemas with the sources and targets used in the data flow.

Note:
If you open a data flow containing old data cleansing transforms (address_enhancement, name_parsing, match_merge), Data Integrator displays the old transforms (even though they no longer appear in the object library). You can even open the properties and see the details for each old transform.

If you attempt to validate a data flow that contains an old data cleansing transform, Data Integrator throws an error. For example:

[Custom Transform:Address_Enhancement] BODI-1116074: First Logic support is obsolete. Please use the new Data Quality feature.

If you attempt to validate a data flow that contains an old data cleansing transform, Data Integrator throws an error. For example:

If you attempt to execute a job that contains data flows using the old data cleansing transforms Data Integrator throws the same type of error.

If you need help migrating your data cleansing data flows to the new Data Quality transforms, contact the SAP Business Objects Help Portal at http://help.sap.com.

6.11.2 Distributed data flows

After upgrading to this version of Data Integrator, existing jobs have the following default values and behaviors:

- Job distribution level: Job.

  All data flows within a job will be run on the same job server.

- The cache type for all data flows: `In-memory` type

  Uses STL map and applies to all join caches, table comparison caches and lookup caches, and so forth.

- Default for **Collect statistics for optimization** and **Collect statistics for monitoring**: deselected.
- Default for **Use collected statistics**: selected.

  Since no statistics are initially collected, Data Integrator will not initially use statistics.

- Every data flow is run as a process (not as a sub data flow process).

New jobs and data flows you create using this version of Data Integrator have the following default values and behaviors:

- Job distribution level: `Job`.
- The cache type for all data flows: Pageable.
- **Collect statistics for optimization** and **Collect statistics for monitoring:** deselected.
- **Use collected statistics:** selected.

If you want Data Integrator to use statistics, you must collect statistics for optimization first.

- Every data flow is run as a single process. To run a data flow as multiple sub data flow processes, you must use the Data_Transfer transform or select the **Run as a separate process** option in transforms or functions.
- All temporary cache files are created under the `<LINK_DIR>\Log\PCache` directory. You can change this option from the Server Manager.

6.11.3 JMS Adapter interface

A new license key may be required to install the JMS Adapter interface. If you have a license key issued prior to Data Integrator XI R2 version 11.5.1, send a request to licensing@businessobjects.com with “Data Integrator License Keys” as the subject line.

6.11.4 XML Schema enhancement

Data Integrator 11.7 adds the new **Include schema location** option for XML target objects. This option is selected by default.

Data Integrator 11.5.2 provided the key XML_Namespace_No_SchemaLocation for section AL_Engine in the Designer option **Tools > Options > Job Server > General**, and the default value, FALSE, indicates that the schema location is included. If you upgrade from 11.5.2 and had set XML_Namespace_No_SchemaLocation to TRUE (indicates that the schema location is NOT included), you must open the XML target in all data flows and clear the **Include schema location** option to keep the old behavior for your XML target objects.

6.11.5 Password management

Data Integrator now encrypts all password fields using two-fish algorithm.

To simplify updating new passwords for the repository database, Data Integrator includes a password file feature. If you do not have a requirement to change the password to the database that hosts the repository, you may not need to use this optional feature.

However, if you must change the password (for example, security requirements stipulate that you must change your password every 90 days), then Business Objects recommends that you migrate your scheduled or external job command files to use this feature.

Migration requires that every job command file be regenerated to use the password file. After migration, when you update the repository password, you need only regenerate the password file. If you do not migrate using the password file feature, then you must regenerate every job command file every time you change the associated password.

6.11.6 Repository size

Due to the multi-byte metadata support, the size of the Data Integrator repository is about two times larger for all database types except Sybase.

6.11.7 Web applications

- The Data Integrator Administrator (formerly called the Web Administrator) and Metadata Reports interfaces have been combined into the new Management Console in Data Integrator 11.7. Now, you can start any Data Integrator Web application from the Management Console launch pad (home page). If you have created a bookmark or favorite that points to the previous Administrator URL, you must update the bookmark to point to http://*computername*:*port*/diAdmin.
- If in a previous version of Data Integrator you generated WSDL for Web service calls, you must regenerate the WSDL because the URL to the Administrator has been changed in Data Integrator 11.7.

6.11.8 Web services

Data Integrator is now using Xerces2 library. When upgrading to 11.7 or above and configuring the Web Services adapter to use the xsdPath parameter in the Web Service configuration file, delete the old Web Services adapter and create a new one. It is no longer necessary to configure the xsdPath parameter.

6.11.9 WSDL version

The Data Integrator web services WSDL has been updated to version 2.0. This version includes changes to address WS-I basic profile compliance. Each message in the WSDL must have at least one part; previously, some messages in the WSDL had no parts. This version also adds new web service

operations for retrieving logging information for batch jobs, retrieving job status, and starting or stopping jobs. These operations have also been made available in older WSDL versions.

Data Quality to Data Services Upgrade

This section provides information about:

- migrating your Data Quality Projects into Data Services
- understanding some of the benefits of using Data Services
- seeing the differences between previous versions of Data Quality and Data Services
- using best practices during migration
- learning how to troubleshoot during migration

7.1 Overview of migration

7.1.1 Data Quality to Data Services Upgrade

This section provides information about:

- migrating your Data Quality Projects into Data Services
- understanding some of the benefits of using Data Services
- seeing the differences between previous versions of Data Quality and Data Services
- using best practices during migration
- learning how to troubleshoot during migration

7.1.2 Who should migrate?

Anyone who is using Data Quality XI and Data Services as standalone applications should migrate to Data Services.

The migration utility works with these versions of software:

- Data Integrator 11.7.x

- Data Quality XI 11.7.x and 11.6.x
- Data Integrator XI R2
- Data Quality XI R2 11.5 and newer[1]
- Firstlogic IQ8 8.05c and newer [2]

Those who are using the Firstlogic Data Quality Suite (Job file, RAPID, Library and/or eDataQuality) cannot use the migration utility to convert the existing projects into Data Services. The only option is to create the projects again in Data Services.

7.1.3 Why migrate?

You may have seen some literature that includes a comprehensive list of reasons to migrate. Here are a handful of the main reasons why you should migrate.

Performance

The new platform utilizes the past Data Integrator features with the improved Data Quality features in one user interface.

Data profiling of source and target

You can monitor, analyze, and report on the quality of information contained in the data marts, data warehouses, and any other data stored in databases. You can test business rules for validity and prioritize data quality issues so that investments can be made in the high impact areas.

Improved multi-user support

With Data Services, you have access to both a central repository (if purchased) for multi-user storage and a local repository for each user. Version control for the repository objects keeps you in control by labeling and comparing objects. This version includes top-notch security with authentication to access the central repository, authorization for group-based permissions to objects, and auditing for changes to each object.

Powerful debugging capabilities

You can set break points, view the data before and after each transform, set filters when previewing data and save and print the preview data.

Repository management

You can easily manage your fully relational repository across systems and have the ability to import repository objects from a file. You can also import source and target metadata for faster UI response times. With datastore configurations, you can define varying connection options for similar datastores between environments, use different relational database technology between environments without changing your jobs, and use different database owners between systems without making changes during migration to each environment. With system configurations, you can associate substitution

[1] Some manual steps are required.
[2] Some manual steps are required.

configurations per system configuration by associating different substitution parameter values by environment.

Import and export metadata

You can import and export metadata with Common Warehouse Model (CWM) 1.0/1.1 support and ERWIN (Computer Associates) 4.x XML. You can also export on Meta Integration Model Bridge (MiMB), if you have it installed.

Auditing

You can audit your projects with statistics collection, rule definitions, email notification, and audit reporting.

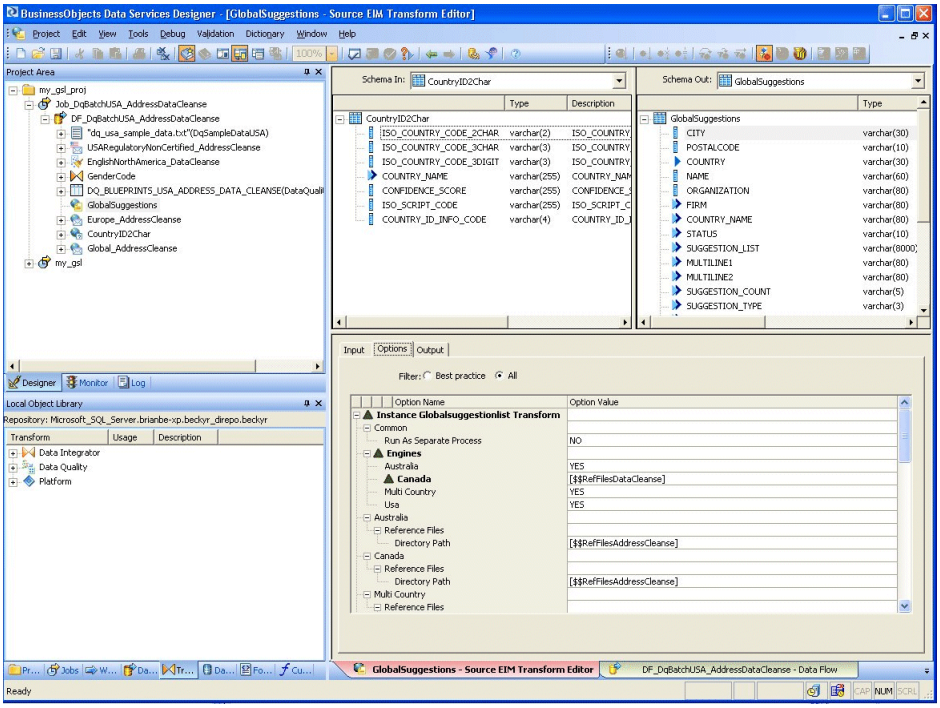
Reports and dashboards

With the reporting tool, you can view daily and historical execution results and duration statistics. With the data validation dashboard, you can view the results of validation rules, organize the validation rules across jobs into functional areas and drill down statistics to a functional area's validation rules. You can also define high-level business rules based on results of validation rules.

With impact and lineage analysis, you can understand the cost and impact of other areas when the datasource is modified. You can view, analyze and print jobs, work flows, and data flow details, view table/file usage based on the source and target, view a summary of job variables and parameters, generate PDF/Word documents on a job-by-job basis, and view a transform option and field mapping summary.

7.1.4 Introduction to the interface

The Data Services user interface is different from the Data Quality user interface. It has similar elements, but in a different presentation.



Note:
The window shows a project named my_gsl_proj open on the left portion of the screen. The right portion shows the GlobalSuggestions transform input and output fields and the Option groups.

In the upper left corner of the Data Services UI, you can see the Project Area. You can see your project folders and any jobs that you have. It's a hierarchical view of the objects used in each project.

Below the Project Area is the is the Local Object Library where you have access to all of the reusable objects in your jobs. It is a view into your repository, so that you do not need to access the repository directly. There are tabs at the bottom so that you can view projects, jobs, work flows, data flows, transforms, data sources, file formats and functions.

The right side of the window is the workspace. The information presented here will differ based on the objects you have selected. For example, when you first open Data Services, you will see a Business Objects banner followed by Getting Started options, Resources, and Recent Projects. In the example, the workspace has the GlobalSuggestions transform open to the object editor. The editor displays the input and output schemas for the object and the panel below lists the options for the object.

See the *Data ServicesDesigner Guide: Designer User Interface* for information.

Related Topics

- Introduction to the interface

7.1.5 Downloading blueprints and other content objects

We have identified a number of common scenarios that you are likely to handle with SAP BusinessObjects Data Services. Instead of creating your own job from scratch, look through the blueprints. If you find one that is closely related to your particular business problem, you can simply use the blueprint and tweak the settings in the transforms for your specific needs.

For each scenario, we have included a blueprint that is already set up to solve the business problem in that scenario. Each blueprint contains the necessary project, jobs, data flows, file formats, sample data, template tables, and custom functions to run the data flows in your environment with only a few modifications.

You can download all of the blueprints or only the blueprints and other content that you find useful from the SAP Community Network website. Here, we periodically post new and updated blueprints, custom functions, best practices, white papers, and other SAP BusinessObjects Data Services content. You can refer to this site frequently for updated content and use the forums to provide us with any questions or requests you may have. We have also provided the ability for you to upload and share any content that you have developed with the rest of the development community.

Instructions for downloading and installing the content objects are also located on the Blueprints web page.

1. To access the SAP BusinessObjects Data Services Blueprints web page, go to http://scn.sap.com/docs/DOC-8820 in your web browser.
2. Open the Content Objects User's Guide to view a list of all of the available blueprints and content objects and their descriptions, and instructions for downloading and setting up the blueprints.
3. Select the blueprint that you want to download.
4. Follow the instructions in the user's guide to download the files to the appropriate location and make the necessary modifications in SAP BusinessObjects Data Services to run the blueprints.

7.1.6 Introduction to the migration utility

The Data Quality Migration Utility is a Windows-based utility command line that migrates your Data Quality repository to the Data Services repository. The utility is in the `<LINK_DIR>\DQMigration` folder. It uses an XML-based configuration file.

You can set options on this Windows-based utility to migrate the entire repository (recommended) or on a project-by-project or transform-by-transform basis. You can also set the utility to Analyze Mode where the utility identifies errors and warning during migration so that you can either fix them in Data Quality before fully migrating.

After running the utility you can optionally view the Migration Report in a web browser for details of possible errors and warnings. We highly recommend you fix these before trying to run the job in Data Services.

In addition, if your Data Quality jobs were published as web services, after running the utility you can publish the migrated jobs as Data Services web services. For information on publishing jobs as web services, see the *Data Services Integrator's Guide*.

Related Topics

- Running the dqmigration utility
- dqmigration utility syntax and options

7.1.7 Terminology in Data Quality and Data Services

Several terms are different between Data Quality and Data Services.

Data Quality	Data Services	Description
folder	project	The terms are different, but they mean the same thing: it holds the project or job that runs.
project	job	In Data Quality, a project is able to run. In Data Services, a project is a level higher. The project contains the job, and the job is able to run.
canvas	workspace	In Data Quality, you dragged a transform onto a canvas. In Data Services, you drag a transform onto a workspace.
dataflow	data flow	In Data Quality, a dataflow is a series of transforms hooked together, that may or may not run. In Data Services, the data flow includes everything that will extract, transform and load data.
option explorer/option editor	object editor	The terms are different, but you do the same things: set your options.
transactional	real time	The terms are different, but they mean the same thing: processing one or many records at a time, usually through a web service.

Data Quality	Data Services	Description
reader	source	The terms are different, but they mean the same thing: a place where the incoming data is held.
writer	target	The terms are different, but they mean the same thing: a place where the output data is held.
substitution variables	substitution parameters	The terms are different, but they mean the same thing: a text string alias.

In Data Quality, you had a few basic layers: a folder, a project, and a dataflow that contains a series of transforms which may or may not run.

In Data Services, the top layer is called a project. The next layer is a job that runs. The job may hold a work flow which is where you can set up conditional processing. The work flow, if you use one, will contain the data flow that contains a series of transforms.

See the *Data Services Designer Guide: Designer User Interface* for information.

7.1.8 Naming conventions

Object names

Your objects, when migrated, will have the prefix DQM_. If the name of the object is longer than 64 characters, then a numeric suffix will be added (for example, _001) to preserve the unique names.

Object	Pre-migrated Data Quality name	Migrated Data Services name
Substitution variable configuration	dqserver1_substitutions.xml	DQM_dqserver1_substitutions.xml
Datastores created from named_connections	first_conn	DQM_first_conn
Datastores created from Reader/Writer settings	Project “first.xml” with Reader called “My_DB_Reader”	DQM_first_My_DB_Reader
File formats created from named_connections	CRM_usa	DQM_CRM_usa

Object	Pre-migrated Data Quality name	Migrated Data Services name
File formats created from Reader/Writer settings	Project “first.xml” with Reader called “Cust_Det”	DQM_first_Cust_Detect
SDK transform configurations	my_usa_reg_addr_cleanse	DQM_my_usa_reg_addr_cleanse
Data Integrator data flows and jobs	first.xml	DQM_first

Input and output field names

Data Quality input and output fields have a period in their names. However, Data Services does not allow a period (.) in the column names. Therefore, the `dqmigration` utility replaces the period with an underscore (_).

For example, suppose your Data Quality Reader transform has input field names `input.field1`, `input.field2`, and `input.field3`. After migration, these field names become `input_field1`, `input_field2`, and `input.field3`.

7.1.9 Deprecated objects

Differences between Data Quality and Data Services

While this list is not exhaustive, it lists the major differences that you will see between Data Quality and Data Services.

There are also changes to transform options and option groups. See each transform section in this document for details.

- Web Services

  The .Net deployment is deprecated.

- Match transforms

  The Aggregator, Sorter[3], Candidate Selector, Match or Associate, Group Statistics, Unique ID, and Best Record transforms are now included together in one Match transform. You will also notice the performance improvement in the Match and UDT transforms.

- GAC transform

  Support for Australia, Canada, Global Address, EMEA, Japan and USA engines. There is also a new Suggestion List option group.

- URAC transform

  You can only output one Suggestion Lists output field within URAC.

[3] The Sorter transform only becomes part of the Match transform when it is migrated in a specific transform order like Sorter, Aggregator and then Match.

- Search/Replace transform

  The Search/Replace Transform is replaced by a Query transform using a new search_replace() function call.

- Data Cleanse transform

  New dictionary connection management window.

- Global Suggestion Lists transform

  New Suggestion List Option group.

- Phonetic Key transform

  The Phonetic Key transform is replaced by a query transform with either a double_metaphone() or soundex() function call.

- Substitution variables and files

  Substitution variables are now referred to as Substitution parameters.

- Python changes

  Several Python methods have been deprecated.

Deprecated objects

The improved technology in Data Services requires the depreciation of certain aspects of Data Quality.

- Compound transforms
- Shared options
- Candidate Selector's Unique ID record stem support
- Some options and behaviors related to pre/post SQL operations in the database Reader and Writer transforms
- UDT's per dataflow mode (use a workflow and scripts as a workaround)
- Disabled transforms

  Note:
  Disabled transforms in Data Quality projects are enabled after migration. If you don't want to the enable the transform, then removed them prior to migration.

- Flat Files
  - Binary file type support in delimited files in versions of Data Quality earlier than 11.7
  - Logical and packed field support in versions of Data Quality earlier than 11.7.
- Data collections
- Thread and watermark settings on a per-transform basis
- Observer transform
- Progress Service transform
- Integrated batch API
- Admin methods in real-time API
- Netscape and Firefox browsers
- JIS_Encoding for flat files
- Some less popular code page support (most used code pages are supported)

- Several Python methods
- Web Services .Net deployment
- Several Web Services functions
- Sun Java application server for web tier

Related Topics

- Introduction to the interface
- Overview of migrated transforms

7.1.10 Premigration checklist

To ensure a smooth migration, make sure you have completed the following tasks.

- Install Data Services.

  Follow the installation instructions in the *Installation Guide.*

- Upgrade to Data Quality XI 11.7, if possible.

  Being on the most current version ensures that the latest options and functionality are properly installed and named. The options will more easily map to the Data Services options.

- Verify that you have permissions to both the Data Quality and Data Services repositories.

  If you don't have a connection to the repositories or permissions to access the repositories, then you will not be able to migrate to Data Services.

- Back up your Data Quality and Data Services repositories.

  Your projects will look different in Data Services. You may want to keep a backup copy of your repository so that you can compare the Data Quality and/or Data Services setup with Data Services.

- Clean your repository.

  Delete any unused projects or projects that have verification errors. Projects that do not run on Data Quality XI will not migrate well and will not run on Data Services without making some changes in Data Services. Projects that have verification errors due to input or output fields will not migrate. Remove any custom transforms, compound transforms and projects that you do not use anymore from the file system repository.

- Verify that your support files are accessible.

  If you have a flat file reader or writer, ensure that the corresponding FMT or DMT file is in the same directory location as the flat file reader or writer.

7.2 Using the migration tool

7.2.1 Overview of the migration utility

You invoke the Data Quality migration utility with the `dqmigration` command and specify a migration configuration file name. The utility is only available on Windows. If your repository is on UNIX, you must have a shared system to the repository, or FTP the repository file to a Windows system prior to running the utility. UNIX customers must run the migration from the Windows client.

The configuration file specifies the Data Quality repository to migrate from, the Data Services repository to migrate to, and processing options for the migration utility. Data Services provides a default migration configuration file named `dqmig.xml` in the directory `<LINK_DIR>\DQMigration`. You can either edit this file or copy it to customize it. For details, see Running the dqmigration utility.

The value of the `LINK_DIR` system variable is the path name of the directory in which you installed Data Services.

The `dqmigration` utility creates the following files:

- Migration report in `<LINK_DIR>\DQMigration\mylogpath`

  This migration report provides the status of each object that was migrated and displays the informational, warning, and error messages from the migration utility. After the `dqmigration` utility completes, it displays a prompt that asks you if you want to view the migration report. Always open the file in Internet Explorer.

- Work files

  The `dqmigration` utility creates a directory `<LINK_DIR>\DQMigration\work_files` that contains the following files. Use these files if you need to troubleshoot any errors in the migration report.

  - Directories configuration_rules_1 and configuration_rules_2 – These directories contain a copy of all the intermediate XML created during migration.
  - .atl files – These files contain the internal language that Data Services uses to define objects.

The last step of the `dqmigration` utility imports the .atl files and creates the equivalent jobs, data flows, and connections in the Data Services repository.

Related Topics

- Running the dqmigration utility
- dqmigration utility syntax and options
- Migration report

7.2.2 Migrating custom Data Quality projects

A Data Quality XI job that contains custom or user-created configurations cannot be migrated to Data Services, without performing manual steps. Because the migration utility cannot recognize these custom configurations created, the manual corrections should be done post migration. You should be able to recognize configurations based on the migration error message you will receive.

You have two options to try:

1. Find the missing configuration and try to create it by looking at the Data Quality XI file. This file should provide you with all of the options for the configuration in the target Data Services repository. After you have created the configuration ATL file, place it in the Data Services repository location admin/repo folder, and then import it into the target repository. When that is done, attempt to migrate the job again.
2. If you do not have success with the first solution, or you do not feel comfortable attempting the first solution, speak to Customer Assurance for help with migrating your custom project.

7.2.3 Migration checklist

During migration, follow these steps.

- Complete the steps in the premigration checklist.
- Run the migration utility on an entire repository. Each repository is upgraded as a unit to ensure that all dependencies for each job are migrated together with the job.
- Follow the utility prompts to complete the migration steps.
- Review the migration report by selecting to view the report at the end of the migration.
- If you have errors or warnings that can be fixed in Data Quality 11.7, fix them and run the utility again. The files for the repository will be overwritten in Data Services when the utility is rerun.
- Fix any other errors or warnings in Data Services.
- Follow the recommendations in each transform section to optimize performance.
- Test the jobs in Data Services and compare the results with Data Quality results.
- Make changes in Data Services, as appropriate.

After you have your jobs migrated to Data Services, you should set aside some time to fully analyze and test your jobs in a pre-production environment.

Related Topics

- Premigration checklist

7.2.4 Connection information

The database connection options may be confusing, especially for <DATABASE_SERVER_NAME> which may be the name of the server, or the name of the database. To set your database connection information, open the dqmig.xml file in the directory <*LINK_DIR*>\DQMigration.

Locate the <DI_REPOSITORY_OPTIONS> section.

Based on your database type, you would enter information similar to the following.

Example: DB2

<DATABASE_TYPE>DB2</DATABASE_TYPE> <!-- Note here that Server name is actually database name-->

<DATABASE_SERVER_NAME>REPORTS1</DATABASE_SERVER_NAME>

<DATABASE_NAME>REPORTS1</DATABASE_NAME>

<WINDOWS_AUTHENTICATION>NO</WINDOWS_AUTHENTICATION>

Example: MS SQL

<DATABASE_TYPE>Microsoft_SQL_Server</DATABASE_TYPE> <!-- Note here that Server name is actual machine if it is default instance.-->

<DATABASE_SERVER_NAME>MACHINE-XP2</DATABASE_SERVER_NAME> <--If you have separate instance then you have to mention something like IQ8TEST-XP2\MSSQL2000 -->

<DATABASE_NAME>REPORTS_MDR</DATABASE_NAME>

<WINDOWS_AUTHENTICATION>NO</WINDOWS_AUTHENTICATION>

Example: MySQL

<DATABASE_TYPE>MySQL</DATABASE_TYPE> <!-- Note here that Server name is actually ODBC DSN name created by installer.-->

<DATABASE_SERVER_NAME>BusinessObjectsEIM</DATABASE_SERVER_NAME>

<DATABASE_NAME>BusinessObjectsEIM</DATABASE_NAME>

<WINDOWS_AUTHENTICATION>NO</WINDOWS_AUTHENTICATION>

Example: Oracle

```
<DATABASE_TYPE>Oracle</DATABASE_TYPE>  <!-- Note here that Server name is actually database name-->

<DATABASE_SERVER_NAME>DSORA103.COMPANY.NET</DATABASE_SERVER_NAME>

<DATABASE_NAME>DSORA103.COMPANY.NET </DATABASE_NAME>

<WINDOWS_AUTHENTICATION>NO</WINDOWS_AUTHENTICATION>
```

Example: Sybase

```
<DATABASE_TYPE>Sybase</DATABASE_TYPE>  <!-- Note here that Server name is actually machine name-->

<DATABASE_SERVER_NAME>sjds003</DATABASE_SERVER_NAME>

<DATABASE_NAME>test_proj </DATABASE_NAME>

<WINDOWS_AUTHENTICATION>NO</WINDOWS_AUTHENTICATION>
```

7.2.5 Running the dqmigration utility

When you run the `dqmigration` utility, you can specify the options in one of the following ways:

- Run the `dqmigration` utility and specify all options from the command line.
- Specify all options in a configuration file. When you run the `dqmigration` utility, enter only the command line option that specifies the name of the configuration file.
- Use a combination of command line options and the configuration file. Values specified from the command line supersede values specified in the configuration file. This allows you to create a configuration file containing default values and then customize as needed from the command line as you migrate each repository. The command line values override the default values in the configuration file.

To run the `dqmigration` utility:

1. Make sure the `PATH` system environment variable contains `%LINK_DIR%/bin`. The utility will not run if it is not there.
2. Set up the configuration file that contains the options for the `dqmigration` utility.

   For example, create a customized configuration file named `dqmig_repo.xml` and copy the contents of the `dqmig.xml` file to the `dqmig_repo.xml` file.
3. Specify the information for the Data Quality repository that you want to migrate.
   a. In the DQ_REPOSITORY_OPTIONS section of the configuration file, specify the following options:

- Absolute path name of your Data Quality repository configuration file in the <CONFIGURATION_RULES_PATH> option.
- Path name, relative to the absolute path name, of your Data Quality substitution file in the <SUBSTITUTION_FILE_NAME> option.

Note:
The migration to SAP BusinessObjects Data Services 3.2 requires that you migrate the entire repository as a complete unit to ensure that each job's dependencies are migrated with the job. Do not enter a value in the <FILE_OR_PATH> option.

b. Specify the information for the Data Services repository to which you want to migrate.

For example, change the options in the `dqmig_repo.xml` configuration file to migrate the Data Quality repository at location `D:\dqxi\11_7\repository\configuration_rules` to the Data Services repository `repo`.

```
<?xml version="1.0" encoding="UTF-16"?>
<REPOSITORY_DOCUMENT FileType = "MigrationConfiguration">
 <MIGRATION_OPTIONS>
  <PROCESSING_OPTIONS>
   <LOG_PATH>mylogpath</LOG_PATH>
   <ANALYZE_ONLY_MODE>YES</ANALYZE_ONLY_MODE>
  </PROCESSING_OPTIONS>
  <DQ_REPOSITORY_OPTIONS>
 <CONFIGURATION_RULES_PATH>D:\dqxi\11_7\repository\configuration_rules</CONFIGURATION_RULES_PATH>
   <SUBSTITUTION_FILE_NAME>dqxiserver1_substitutions.xml</SUBSTITUTION_FILE_NAME>
   <FILE_OR_PATH></FILE_OR_PATH>
  </DQ_REPOSITORY_OPTIONS>
  <DI_REPOSITORY_OPTIONS>
   <DATABASE_TYPE>Microsoft_SQL_Server</DATABASE_TYPE>
   <DATABASE_SERVER_NAME>my_computer</DATABASE_SERVER_NAME>
   <DATABASE_NAME>repo</DATABASE_NAME>
   <WINDOWS_AUTHENTICATION>NO</WINDOWS_AUTHENTICATION>
         <USER_NAME>repo</USER_NAME>
         <PASSWORD>repo</PASSWORD>
  </DI_REPOSITORY_OPTIONS>
 </MIGRATION_OPTIONS>
</REPOSITORY_DOCUMENT>
```

4. Open a command window and change directory to *<LINK_DIR>*`\DQMigration` where the `dqmigration` executable is located.
5. Run the `dqmigration` utility in analyze mode first to determine the status of objects within the Data Quality repository.

   Specify the value `YES` for <ANALYZE_ONLY_MODE> in the configuration file, or use the option -a when you run the dqmigration utility. For example, type the following command and options in a command window:

   ```
   dqmigration -cdqmig_repo.xml -aYES
   ```

   Note:
   There is no space between the option -a and the value YES .

   This step does not update the Data Services repository.
6. When the migration utility displays the prompt that asks if you want to view the migration report, reply `y`.
7. In the migration report, review any messages that have type `Error`.
   - You should migrate a production Data Quality repository where all of the projects have executed successfully. If your Data Quality repository contains unverifiable or non-runnable projects, then

they will likely not migrate successfully. Delete these types of projects before you run the `dqmigration` utility with <ANALYZE_ONLY_MODE> set to `NO`.

Note:
To delete a project in Data Quality, you delete its .xml file from the directory `\repository\configuration_rules\projects`.

- If your Data Quality repository contains objects that the migration utility currently does not migrate (for example dBase3 data sources), you need to take some actions before you migrate. For details, see Troubleshooting.

Other errors, such as a Reader or Writer that contains a connection string, can be fixed in Data Services after you run the migration utility with <ANALYZE_ONLY_MODE> set to `NO`.

For information about how to fix errors listed in the migration report, see Troubleshooting.

8. After you have fixed the serious errors that pertain to the Data Quality projects that you want to migrate, run the `dqmigration` utility with <ANALYZE_ONLY_MODE> set to `NO`.

   This step imports the migrated objects into the Data Services repository.

Some of the Data Services jobs and data flows that result from the migration from Data Quality might require clean-up tasks before they can execute. For details, see Further cleanup .

Other migrated jobs or data flows might require changes to improve their performance. For details, see Improving performance .

7.2.6 dqmigration utility syntax and options

Use the `dqmigration` utility to migrate the contents of a Data Quality repository to a Data Services repository.

If you specify an option in the command line, it overrides the value in the configuration file.

Note:
If your configuration file is invalid, the command line options will not be processed.

The following table describes the `dqmigration` utility options.

Option	XML tag in dqmig configuration file	Description
`-h` or `/h` or `/?`	None	Prints the options available for this utility.
`-cmig_conffile_pathname`	None	Name of the configuration file for this utility. Default value is `dqmig.xml` which is in the *<LINK_DIR>*`\DQMigration` directory.

Option	XML tag in dqmig configuration file	Description
-l*mig_logfile_path*	<LOG_PATH> *mig_logfile_path* </LOG_PATH>	Path for the log file that this dqmigration utility generates. Default value is `migration_logs` which is in the current directory from which you run this utility.
-a*mode*	<ANALYZE_ONLY_MODE> *mode* </ANALYZE_ONLY_MODE>	Analyze mode. Specify `YES` to analyze the Data Quality repository and provide warning and error messages for objects that will require manual steps to complete the migration, but do not update the Data Services repository. Specify `NO` to also update the Data Services repository with the migrated objects. Default value is `NO`.
-t*timeout_in_seconds*	<IMPORT_TIMEOUT> *timeout_in_seconds* </IMPORT_TIMEOUT>	Amount of time that the migration utility waits to import one Data Quality project into the Data Services repository. If one project times out, the migration utility will continue with the next project. If a timeout occurs, the migration log indicates which object was being processed. Default value is 300 seconds.
-ri*dq_input_repos_path*	<CONFIGURATION_RULES_PATH> *dq_input_repos_path* </CONFIGURATION_RULES_PATH>	Path of the Data Quality repository files. Default value is blank, but the sample has the value `dq_directory\repository\configuration_rules` where *dq_directory* is the directory in which you installed Data Quality.
-rs*dq_substitution_file*	<SUBSTITUTION_FILE_NAME> *dq_substitution_file* </SUBSTITUTION_FILE_NAME>	Name of the substitution file to use during migration. Default value is blank, but the sample has the value `dqxiserver1_substitutions.xml`

Option	XML tag in dqmig configuration file	Description
-fi*dq_input_repos_file*	<FILE_OR_PATH> *dq_input_repos_file* </FILE_OR_PATH>	Default value is empty (no value specified). The entire Data Quality repository is migrated. Accept the default value. Prior to SAP BusinessObjects Data Services 3.2 this option was used to specify the name of an XML file or folder to migrate. The option is no longer supported.
-dt*ds_repo_db_type*	<DATABASE_TYPE> *ds_repo_db_type* </DATABASE_TYPE>	Database type for Data Services repository. Values can be one of the following: • ODBC • Oracle • Microsoft_SQL_Server • Sybase • MySQL • DB2
-ds*di_repo_db_server*	<DATABASE_SERVER_NAME> *di_repo_db_server* </DATABASE_SERVER_NAME>	Database server name for a Data Services repository that is Microsoft SQL Server or Sybase ASE.
-dn*di_repo_db_name*	<DATABASE_NAME> *di_repo_db_name* </DATABASE_NAME>	Connection name for a Data Services repository that is Oracle. Data source name for a Data Services repository that is DB2 or MySQL. Database name for a Data Services repository that is Microsoft SQL Server or Sybase ASE.
-dw*win_auth*	<WINDOWS_AUTHENTICATION> *win_auth* </WINDOWS_AUTHENTICATION>	Specifies whether to use Windows Authentication (instead of SQL Server authentication) for Microsoft SQL Server. Values can be Yes or No. The default is No.
-du*di_repo_user_name*	<USER_NAME> *di_repo_user_name* </USER_NAME>	User name of authorized user to the Data Services repository.
-dp*di_repo_password*	<PASSWORD> *di_repo_password* </PASSWORD>	Password for the user authorized to the Data Services repository.

7.2.7 Migration report

When you run the `dqmigration` utility, it generates a migration report in your `<LINK_DIR>\DQMigration\mylogpath` directory.

The `dqmigration` utility displays the name of the migration report and a prompt that asks if you want to view it. The dqmigration log file has a name with the following format:

```
dqmig_log_processID_threadID_yyyymmdd_hhmmssmmm.xml
```

For example: `dqmig_log_005340_007448_20080619_092451642.xml`

The migration report consists of the following sections:

- Migration Summary
  - Migration Settings – Lists the configuration option values used for this migration run.
  - General Processing Messages – Lists the status message of every .xml file processed in the Data Quality repository. Possible message types are:
    - `Info` when the .xml file was processed successfully.
    - `Error` when the .xml file was not processed successfully, and you will need to take corrective action.
    - `Warning` when the .xml file was processed and migrated, but you might need to take additional actions after migration to make the resulting data flow run successfully in Data Services. For example, a substitution variable is not migrated and you would need to enter the actual value in the Data Services object after migration.

    The `dqmigration` utility executes in the following three stages, and this General Processing section displays a different color background for the messages that are issued by each stage.
    - Stage 1 — Determines the version of the source Data Quality repository. If the Data Quality repository is a version prior to 11.7, this stage migrates it to version 11.7. The background color of these stage 1 messages is beige.
    - Stage 2 — Migrates Global Address Cleansing and Match family transforms. The background color of these stage 2 messages is light pink.
    - Stage 3 — Migrates all other Data Quality objects and creates the .atl file to import into the Data Services repository. The background color of these stage 3 messages is the background color set for your Web browser.
  - General Objects – Lists the status of substitution files and connection files.
  - Transform Configurations – Lists the status of each transform configuration that was migrated.
  - Jobs – Lists the status of each job that was migrated.
- Migration Details

  This section provides detailed migration status for each of the following objects:
  - Each substitution file
  - Each named connection in the named_connections.xml file.

- Each transform configuration
- Each job (project)

7.3 How Data Quality repository contents migrate

This section describes how the `dqmigration` utility migrates the following Data Quality Repository components which consist mainly of different XML files and folders:

- Projects and folders
- Substitution files and variables
- Named connections
- Data types
- Content type

Related Topics

- Deprecated objects

7.3.1 How projects and folders migrate

In Data Quality, folders contain objects, and the folder hierarchy can be multiple levels. After migration to Data Services, the utility prefixes each object with `DQM_`, and then they appear together in the Local Object Library. The "Properties" of the migrated object indicates its source Data Quality project.

Note:
All of the files are not migrated. For example, the contents of the blueprints folder, and the sample transforms that are automatically installed in Data Quality. If you want to migrate a file from that folder, move the contents to another folder that is migrated.

In Data Quality, you can create three types of projects:

- Batch – A batch project executes at a specific time and ends after all data in the specified source is processed. See How batch projects migrate.
- Transactional – A transactional project receives data from a calling application (such as Web services or SDK calls) and processes data as it arrives. It remains active and ready for new requests. See How transactional projects migrate.
- Integrated Batch -- A project run using batch processing with an Integrated Batch Reader and an Integrated Batch Writer transform. This type of project can be used to pass data to and from an integrated application, including Data Integrator XI Release 2 Accelerated (11.7). See How integrated batch projects migrate.

Related Topics

- How database Reader transforms migrate
- How database Writer transforms migrate

- How flat file Reader transforms migrate
- How flat file Writer transforms migrate
- How transactional Readers and Writers migrate

7.3.1.1 How batch projects migrate

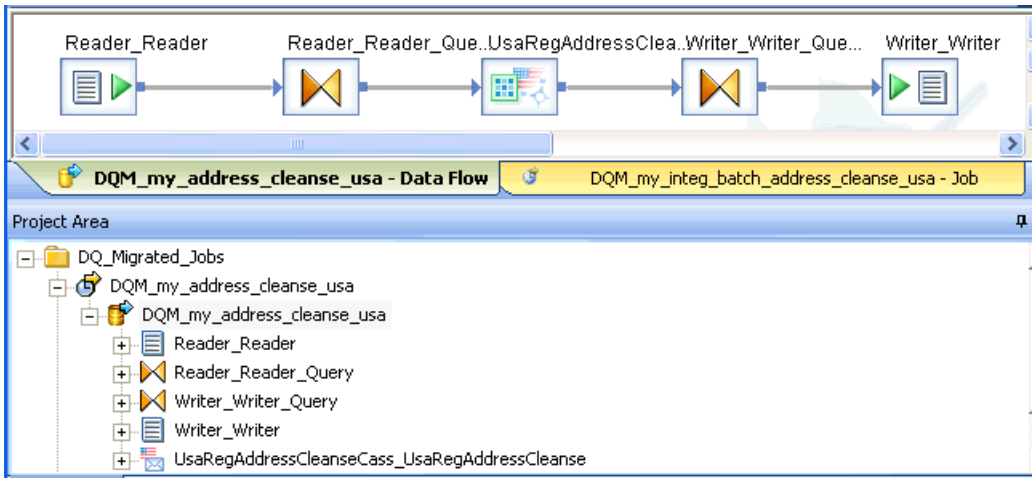
The `dqmigration` utility migrates a Data Quality Batch project to the following two Data Services objects:

- Data Flow: A Data Services“data flow” is the equivalent of a Data Quality project. Everything having to do with data, including reading sources, transforming data, and loading targets, occurs inside a data flow.
- Job: A “job” is the only object you can execute in Data Services. Therefore, the migration utility creates a job to encompass the migrated data flow.

The `dqmigration` utility also creates additional objects, such as a datastore or file format, as sections How database Reader transforms migrate and How flat file Reader transforms migrate describe.

For example, suppose you have the following Data Quality my_address_cleanse_usa project.

The `dqmigration` utility migrates this batch project to a Data Services job named DQM_my_address_cleanse_usa which contains a data flow with the same name. The `dqmigration` utility adds a `DQM_` prefix to your project name from Data Quality to form the Data Services job and data flow name.

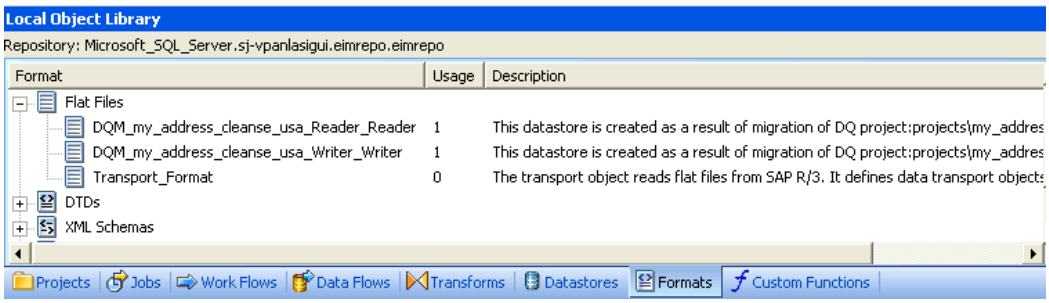


The Project Area shows that the data flow DQM_my_address_cleanse_usa contains the following migrated transforms:

- The Data Quality Reader transform has become the following Data Services objects:
  - Source object named `Reader_Reader`, which reads data from file format DQM_my_address_cleanse_usa_Reader_Reader. The file format is visible on the Formats tab of the Local Object Library.
  - Query transform named `Reader_Reader_Query`, which maps fields from the file format to the field names used by the rest of the data flow.
- The Data Quality UsaRegAddressCleanseCass transform has become the Data Services UsaRegAddressCleanseCass_UsaRegAddressCleanse transform.
- The Data Quality Writer transform has become the following Data Services objects:
  - Query transform named `Writer_Writer_Query`
  - Target object named `Writer_Writer`, which maps fields from field names used in the data flow to the file format `DQM_my_address_cleanse_usa_Writer_Writer` (visible in the Local Object Library).

When the `dqmigration` utility creates the Data Services objects, it updates each object's description to indicate its source Data Quality project and transform. The description also contains the original Data Quality description.

For example, in the Data ServicesDesigner, the Formats tab in the Local Object Library shows the following descriptions.



To view the full description, select a name in the Format list, right-click, and select **Properties**.

Related Topics

- How database Reader transforms migrate
- How database Writer transforms migrate

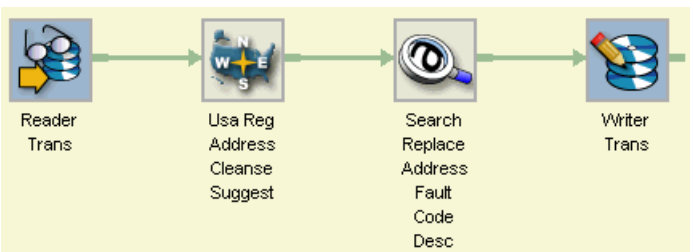
- How flat file Reader transforms migrate
- How flat file Writer transforms migrate

7.3.1.2 How transactional projects migrate

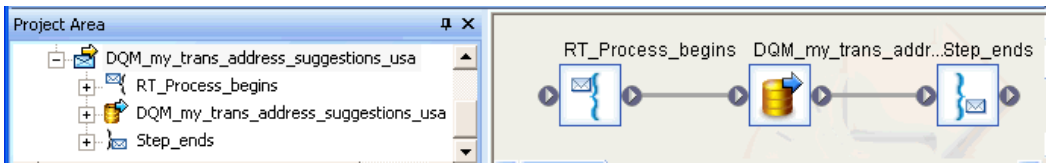
The `dqmigration` utility migrates a Data Quality transactional project to a Data Services real-time job that consists of the following objects:

- Initialization: Starts the real-time process. The initialization component can be a script, work flow, data flow, or a combination of objects. It runs only when a real-time service starts.
- A data flow which contains the resulting Data Services objects from the Data Quality project:
  - A single real-time source: XML message
  - A single real-time target: XML message
- Clean-up: Ends the real-time process. The clean-up component (optional) can be a script, work flow, data flow, or a combination of objects. It runs only when a real-time service is shut down.

For example, suppose you have the following Data Quality project my_trans_address_suggestions_usa.xml.



The `dqmigration` utility migrates this transactional project my_trans_address_suggestions_usa to a Data Services real-time job named DQM_my_trans_address_suggestions_usa which contains a data flow with the same name.



Related Topics

- How transactional Readers and Writers migrate
- Post-migration tasks for transactional Readers and Writers

7.3.1.3 How integrated batch projects migrate

In Data Integrator XI Release 2 Accelerated (version 11.7), a Data Quality integrated batch project passed data from a job in Data Integrator, cleansed the data in Data Quality, and passed the data back to the Data Integrator job. These Data Quality projects were imported into Data Integrator 11.7 through a Data Quality datastore and the imported Data Quality project was used as a cleansing transform within a Data Integrator data flow.

The Data Quality transforms are now integrated into Data Services, and you can use them just like the regular Data Integrator transforms in a data flow. You no longer need to create a Data Quality datastore and import the integrated batch projects.

If you used imported Data Quality projects in Data Integrator 11.7, you will see them in Data Services.

- Data Quality datastores are still visible, but you cannot edit them and you cannot create new ones. You can delete the Data Quality datastores.
- You cannot connect to the Data Quality Server and browse the integrated batch projects.
- The previously imported Data Quality projects are still visible:
  - under each Data Quality datastore, but you cannot drag and drop them into data flows. You can use the option **View Where Used** to see what existing data flows use each imported Data Quality project.
  - as a Data Quality transform within the data flow, and you can open this transform to view the field mappings and options.

To use your existing Data Integrator 11.7 integrated batch projects in Data Services, you must modify them in one of the following ways:

- If your Data Integrator 11.7 data flow contains a small number of imported Data Quality transforms, modify your data flow in Data Services to use the new built-in Data Quality transforms.

- If your Data Integrator 11.7 data flow contains a large number of Data Quality transforms, migrate the Data Quality integrated batch project and replace the resulting placeholder Readers and Writers with the appropriate sources and targets. A large number of Data Quality transforms can exist as either:
  - An imported Data Quality project that contains a large number of Data Quality transforms.
  - Multiple imported Data Quality projects that each contain a few Data Quality transforms.

Related Topics

- Modifying Data Integrator 11.7 Data Quality projects
- Migrating Data Quality integrated batch projects
- How batch projects migrate

7.3.2 How connections migrate

The `dqmigration` utility migrates Data Quality connections to one of the following Data Services objects, depending on whether the data source is a file or a database:

- Datastores when the connection is a database -- “Datastores” represent connection configurations between Data Services and databases or applications.
- File Formats when the connection is a flat file -- A “file format” is a set of properties describing the structure of a flat file (ASCII). File formats describe the metadata structure, which can be fixed or delimited. You can use one file format to access multiple files if they all have the same metadata structure.

For a named connection, the `dqmigration` utility generates the name of the resulting datastore or file format as follows:

```
DQM_connectionname
```

For connections that are not named (the Reader or Writer transform provides all of the connection information instead of the file `named_connections.xml`), the resulting name of the datastore or file format is:

```
DQM_projectname_readername_Reader
```

```
DQM_projectname_writername_Writer
```

The following table shows to which Data Services objects each specific Driver_Name value is migrated.

Table 7-2: Mapping of Data Quality Driver_Name options

Value of Data Quality connection Driver_Name option	Data Services object
FLDB_DBASE3	Placeholder Fixed width File Format

Value of Data Quality connection Driver_Name option	Data Services object
FLDB_STD_DB2	Datastore
FLDB_DELIMITED	File Format for a flat file
FLDB_FIXED_ASCII	File Format for a flat file
FLDB_STD_MSSQL	ODBC Datastore
FLDB_STD_MYSQL	Placeholder ODBC Datastore
FLDB_STD_ODBC	Datastore
FLDB_STD_ORACLE9	Datastore
FLDB_STD_ORACLE10	Datastore

Rather than not migrate the Data Quality xml file, the `dqmigration` utility creates a placeholder file format or placeholder datastore that you can fix after migration in situations that include the following:

- The flat file type or database type (such as dBase3) is not supported in Data Services.
- A substitution variable was used in the Data_Source_Name and the file that the substitution variable refers to was not found.

For information about how to change the placeholder objects, see *Connection Errors* in Troubleshooting.

The following table shows how the Data Quality connection options are mapped to the Data Services datastore options.

Note:
If the `dqmigration` utility cannot determine the database type from a named connection (for example, the connection name specified in the Data Quality Reader does not exist), then the utility creates an Oracle 9 datastore.

Table 7-3: Connection_Options mapping

Data Quality Connection_Options option	Data Services datastore option
Named_Connection	**Datastore name**

Data Quality Connection_Options option		Data Services datastore option
Data_Source_ Name	FLDB_STD_DB2	**Data Source**
	FLDB_STD_MSSQL	**Data Source**
	FLDB_STD_MYSQL	Cleanup required. See Fixing invalid datastores.
	FLDB_STD_ODBC	**Data Source**
	FLDB_STD_ORACLE9	**Connection name**
	FLDB_STD_ORACLE10	**Connection name**
	Connection string	Cleanup required. See Fixing invalid datastores.
Driver_Name	FLDB_STD_DB2	**Database type:**`DB2` **Database version:**`DB2 UDB 8.x`
	FLDB_STD_MSSQL	**Database type:**`ODBC`
	FLDB_STD_MYSQL	A placeholder ODBC datastore is created. Cleanup required. See Fixing invalid datastores.
	FLDB_STD_ODBC	**Database type:**`ODBC` *`Driver_name`* in **ODBC Data Source Administration**
	FLDB_STD_ORACLE9	**Database type:**`Oracle` **Database version:**`Oracle 9i`
	FLDB_STD_ORACLE10	**Database type:**`Oracle` **Database version:**`Oracle 10g`
Host_Name		Ignored, but the `dqmigration` log displays a warning.
PORT_NUMBER		Ignored, but the `dqmigration` log displays a warning.
User_Name		In Edit Datastore, **User name**
Password		In Edit Datastore, **Password**

For example, suppose your Data Quality projects use the following named connections:

Existing data source connections:

Name	Context	Driver Name	Host Name	Data Source Name	Port Number	User Name
namedoutff	Client	FLDB_FIXED_ASCII		D:\dq_files\outff.txt	0	
namedoutff	Server	FLDB_FIXED_ASCII		D:\dq_files\outff.txt	0	
orcl	Client	FLDB_STD_ORACLE10		orcl	0	myred
orcl	Server	FLDB_STD_ORACLE10		orcl	0	myred

The following Formats tab in the local object library in the Data Services Designer shows:

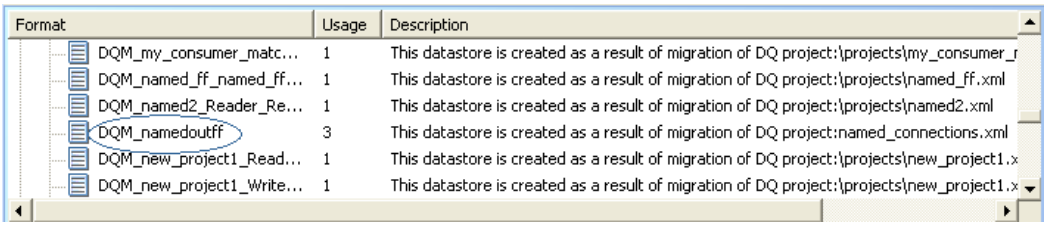
- The Data Quality named connection `namedoutff` is now the Data Services file format `DQM_namedoutff`.

  Note:
  A Data Quality named connection can have two contexts: Client and Server. The `dqmigration` utility migrates the Server named connection and ignores the Client one.

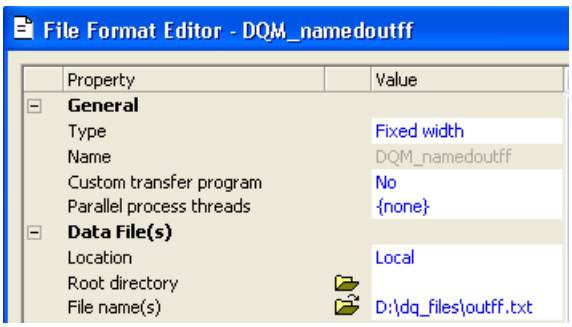
  For a named connection, the description of the resulting file format refers to named_connections.xml.

- Other Data Quality connections migrated to file formats in addition to the `namedoutff` flat file connection.

  These other Reader and Writers do not use named connections; therefore, the description indicates the name of the Data Quality project from which it was created.



To view the connection information, select a name in the Format list, right-click, and select **Edit**.



- The Data Quality**Driver Name** `FLDB_FIXED_ASCII` becomes the Data Services File Format **Type** `Fixed width`.
- The Data Quality connection **Name** `namedoutff` becomes the Data Services File Format **Name** `DQM_namedoutff`.

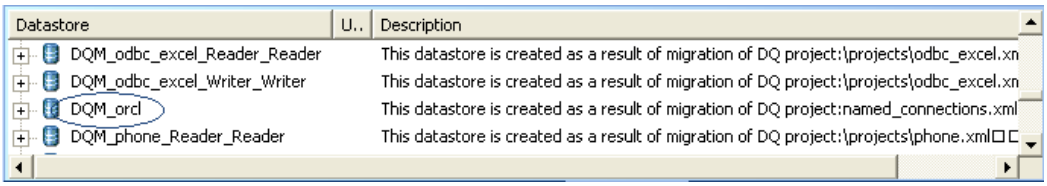
- The Data Quality**Data Source Name** `D:\dq_files\outff.txt` becomes the Data Services**File name(s)**.

The following Datastores tab in the local object library in the Data Services Designer shows:

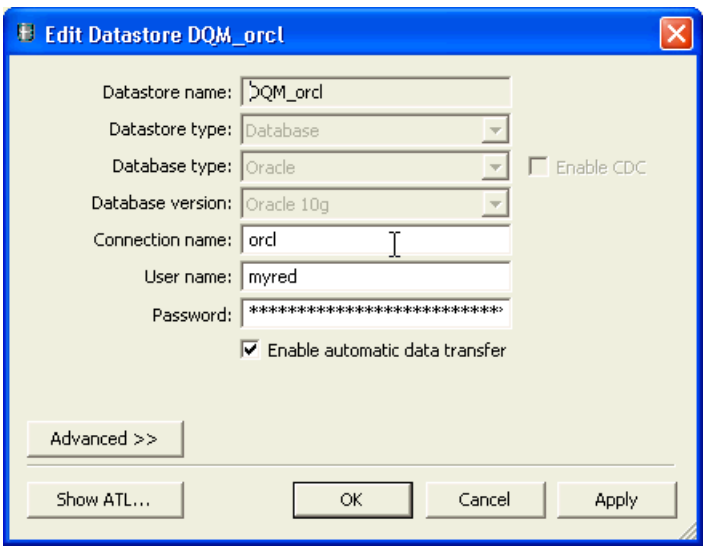
- The Data Quality named connection `orcl` is now the Data Services datastore `DQM_orcl`.

  For a named connection, the description of the resulting datastore refers to named_connections.xml. Otherwise, the description indicates the name of the Data Quality project from which it was created.

- Other Data Quality connections migrated to datastores in addition to the `orcl` database connection.



To view the connection information, select a datastore name in the datastore list, right-click, and select **Edit** .



- The Data Quality connection **Name** `orcl` becomes the Data Services**Datastore name** `DQM_orcl`.
- The Data Quality**Data Source Name** `orcl` becomes the Data Services**Connection name** `orcl`.

7.3.3 How substitution files and variables migrate

The `dqmigration` utility migrates Data Quality substitution files and variables as follows :

- Each Data Quality substitution file becomes a Data Services substitution configuration in the Substitution Parameter Store. The name of the resulting substitution configuration has the following format:

  ```
  DQM_substitutionfilename
  ```

- Each Data Quality substitution variable becomes a Data Services substitution parameter. The name of the resulting substitution parameter is the same as the substitution variable. After migration, the substitution parameter is enclosed in square brackets ([]) when used in the data flow.

After migration, you need to take additional steps to ensure that the migrated substitution parameters use the correct values. For details, see Setting substitution parameter configuration.

Note:
In Data Quality, the substitution variables are case-sensitive. However, in Data Services, the substitution parameters are case-insensitive. Therefore, if two substitution variables exist with the same name but different casing, only the first substitution variable migrates and the `dqmigration` utility issues a warning message in the migration report.

Related Topics

- Name differences with substitution parameters for reference files
- How built-in substitution variables migrate
- Example of how substitution files and variables migrate

7.3.3.1 How built-in substitution variables migrate

The following table shows that $$Dataflow_Name is the only Data Quality 11.7 built-in substitution variable that migrates to Data Services . The `dqmigration` utility replaces all instances of $$Dataflow_Name with the Data Quality project name.

Data Quality built-in substitution variable	Migrate to Data Services
$$Dataflow_Name	Yes
$$Dataflow_XML_Path	No
$$Log_Path	No
$$Report_Path	No
$$Stat_Path	No
$$Time_Stamp	No
$$Dataflow_ID	No

Data Quality built-in substitution variable	Migrate to Data Services
$$Runtime_Metadata_Path	No
$$Repos_Path	No

The `dqmigration` utility leaves the unsupported built-in substitution variables as they are in the migrated data flow and issues warning messages in the Migration Report. You must change these substitution variables to the actual values in the Data Services data flow. If you do not change them, Data Services produces run-time errors.

Note:
If any of these built-in substitution variables are in places that the ATL syntax does not allow substitution variables, the import of the generated .atl file will fail. You must take one of the following actions before you migrate the project:

- Delete or change these variables to the actual values.
- Remove the references to these variables.

When $$Dataflow_Name is used in a flat file Reader transform, the `dqmigration` utility does the following:

- Replaces the substitution variable with the actual value.
- Locates the .fmt or .dmt file with the dataflow name.
- Creates a corresponding file format.

However, $$Dataflow_Name is no longer a substitution variable. Therefore, if the name of the dataflow changed, you must change the actual value to avoid unexpected results.

7.3.3.2 Name differences with substitution parameters for reference files

The installation of Data Services creates a substitution configuration that includes substitution parameters for Data Quality reference files used by various content transforms. Most of the Data Quality substitution variables have been replaced by a single Data Services substitution parameter. These substitution parameter settings are now part of the base configurations for various content transforms.

The Data Services names of these substitution parameters differ from the names of the corresponding substitution variables in Data Quality as the following table shows.

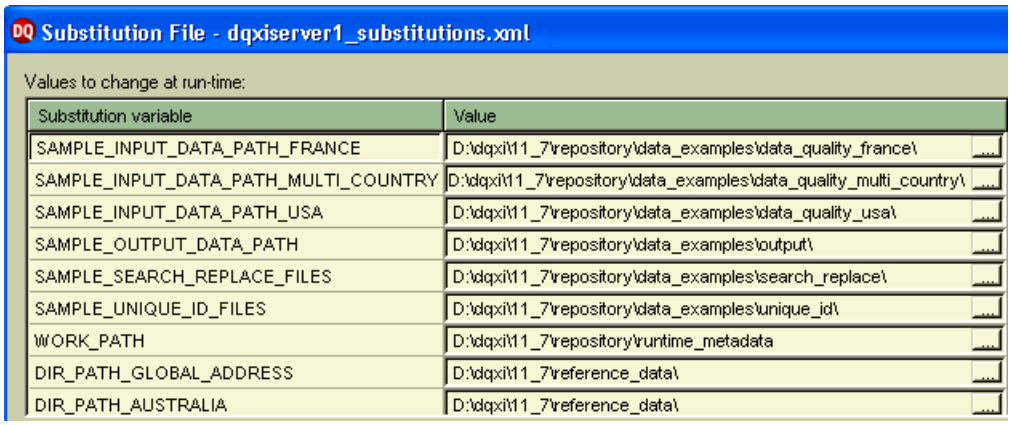
Table 7-5: Name differences for substitution parameters for reference files

Data Quality substitution variable name	Data Services substitution parameter name
$$DIR_PATH_GLOBAL_ADDRESS	$$RefFilesAddressCleanse
$$DIR_PATH_AUSTRALIA	
$$DIR_PATH_CANADA	
$$DIR_PATH_JAPAN	
$$DIR_PATH_MULTI_COUNTRY	
$$DIR_PATH_USA	
$$DIR_PATH_USA_DPV	
$$DIR_PATH_USA_ELOT	
$$DIR_PATH_USA_EWS	
$$DIR_PATH_USA_GEO	
$$DIR_PATH_USA_LACS	
$$DIR_PATH_USA_RDI	
$$DIR_PATH_USA_Z4CHANGE	
Part of $$DCT_PATH	$$RefFilesDataCleanse
Did not exist in Data Quality	$$ReportsAddressCleanse
Did not exist in Data Quality	$$ReportsMatch
Did not exist in Data Quality	$$SamplesInstall

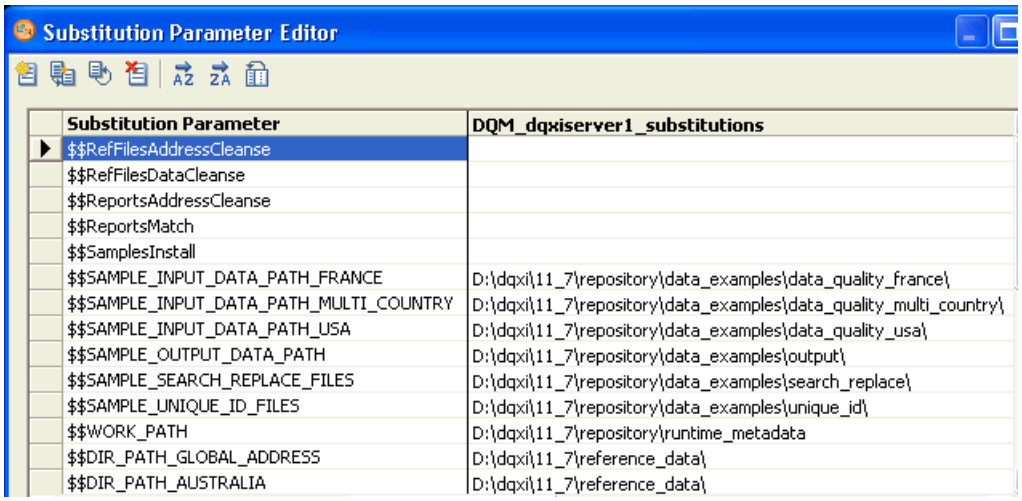
Note:
Ensure that you install the directories and other reference files in the path pointed to by these substitution parameters.

7.3.3.3 Example of how substitution files and variables migrate

Suppose you have the following Data Quality substitution file "dqxiserver1_substitutions" that contains the following substitution variables.



`dqmigration` utility migrates this Data Quality substitution file and variables to the following Data Services substitution configuration.



7.3.4 How data types migrate

The following table shows how the `dqmigration` utility maps the Data Quality field types to the Data Services data types.

Table 7-6: Mapping of data types

Data Quality data type	Data Services data type
NUMBER	decimal (38,7)

Data Quality data type	Data Services data type
DOUBLE	double
DATE	date
DATETIME	datetime
TIME	time
INT, UNIT	decimal
STRING	varchar
BINARY	blob in fixed-width files
BYTEARRAY	varchar

Note:
Data Quality does not distinguish between an integer and a decimal. It represents and stores both data types as `number`. However, Data Services makes this distinction. The `dqmigration` utility converts a number data type to decimal(38,7) which accommodates both decimal and integer values. After migration, verify that this decimal(38,7) is appropriate for your data. If it is not, change the data type.

7.3.5 How Data Quality attributes migrate

The Data_Record_Field_Type and Data_Record_Field_Length are automatically mapped in each transforms' Schema Out Type column.

Likewise, the Content_Type attribute is automatically mapped in each transform's Schema Out Content Type column. If a content type cannot be automatically mapped, then the content type will be blank.

You can change the attribute values for an output field by right-clicking the field and selecting Properties.

7.4 How transforms migrate

7.4.1 Overview of migrated transforms

This topic lists the Data Quality transforms, how Data Services implements each transform after it is migrated, and the new equivalent Data Services transform if you create a new Data Services data flow.

Data Quality transform	Data Services objects after migration	Equivalent Data Services transform
Aggregator	Match transform	Part of the Group Forming pre-match operations in the Match transform. (Break Group)
Associate	Associate transform	
Best Record	Match transform	Part of the match-level post-match processing operations.
Candidate Selector	Match transform.	Part of the Group Forming pre-match operations in the Match transform. (Candidate Selection)
Copier	Query transform with multiple output pipes.	Built-in functionality
Country_id	How Country ID transforms migrate	See the Country_ID transform family table below.
Data Cleanse	Similar to Data Cleanse in Data Quality.	See the Data Cleanse transform family table below.
Filter	Python-based User-Defined and Case transform to support multiple pipes	Case transform
Formatter	User-Defined transform	Query transform. See Formatter transform
Global Address Cleanse	How Global Address Cleanse transforms migrate	See the Global Address Cleanse transform family table below.
Global Suggestion Lists	Similar to Global Suggestion Lists in Data Quality.	See the Global Suggestion Lists transform family table below.
Group Statistics	Match transform.	Group Statistics post-match processing per match level.
List/Source	User-Defined transform	Query transform
Match	Match transform	See the Match transform family (non-priority) table below.
Observer	Query transform	Not supported because this transform does not exist in Data Quality 11.5 and above.
Phonetic Key	Query with Soundex or Double-Metaphone function	No longer a transform.

Data Quality transform		Data Services objects after migration	Equivalent Data Services transform
Progress Service		Not migrated	Not supported
Reader	Database Reader	Database source and SQL transform with Query. See How database Reader transforms migrate.	Database source imported from database datastore
	Flat File Reader	Flat file source and Query transform. See How flat file Reader transforms migrate.	Flat file source imported from file format
	Transactional Reader	XML message source, Query_Reader transform, Row_Generation transform, Query_Driver transform. See How transactional Readers and Writers migrate.	XML message source imported from file format
Scan and Split		User Defined transform with scan-split Python functions	Query transform with replace_substr function
Search and Replace		Search_replace function in Query transform	Query transform with search_replace function
Sorter		Order By option in Query transform. In some cases, part of Match transform options.	Order By option in Query transform
Unique ID		Match transform.	Part of the match-level post-match processing operations.
Unique ID Plugin		Match transform.	Part of the Unique ID post-match processing operation.
USA Regulatory Address Cleanse		How USA Regulatory Address Cleanse transforms migrate	See the USA Regulatory Address Cleanse transform family table below.
User-Defined transform		User-Defined transform	See How User-Defined transforms migrate
Writer	Database Writer	Query transform and dummy target table. See How database Writer transforms migrate.	Database target imported or created from database datastore
	Flat File Writer	Query transform and flat file target. See How flat file Writer transforms migrate.	Flat file source imported or created from file format
	Transactional Writer	Query transform and XML schema target. See How transactional Readers and Writers migrate.	XML message source imported or created from file format

The following section discusses each transform family and maps the Data Quality transform to the equivalent Data Services transform.

Note:
The migration utility only processes transforms that are created by the user. If a transform was part of the Data Quality installation, it will not be migrated because the equivalent transform will be installed as part of the Data Services installation.

Table 7-8: Associate transform family

Data Quality sample transform	Equivalent Data Services transform configuration
assoc_match_sets	AssociateGroupStatistics_AssociateBatch
assoc_trans	Base_Associate
associate	Wizard_AssociateBatch

Table 7-9: Country_ID transform family

Data Quality sample transform	Equivalent Data Services transform configuration
country_id_country_ code	CountryID2Char

Table 7-10: Data Cleanse transform family

Data Quality sample transform	Equivalent Data Services transform configuration
data_cleanse_en	EnglishNorthAmerica_DataCleanse
data_cleanse_es	Spanish_DataCleanse
data_cleanse_fr	French_DataCleanse
data_cleanse_pt	Portuguese_DataCleanse

Table 7-11: Global Address Cleanse transform family

Data Quality sample transform	Equivalent Data Services transform configuration
australia_plugin	Global_AddressCleanse **Note:** The Australia plugin is now the Australia engine within the Global Address Cleanse Transform.
canada_plugin	Global_AddressCleanse **Note:** The Canada plugin is now the Canada engine within the Global Address Cleanse Transform.
global_address_ cleanse	Global_AddressCleanse
global_address_ cleanse_australia	Australia_AddressCleanse
global_address_ cleanse_brazil	Brazil_AddressCleanse
global_address_ cleanse_canada	Canada_AddressCleanse
global_address_ cleanse_canada_ suggest	Canada_AddressCleanse
global_address_ cleanse_france	France_AddressCleanse
global_address_ cleanse_germany	Germany_AddressCleanse
global_address_ cleanse_italy	Italy_AddressCleanse
global_address_ cleanse_japan	Japan_AddressCleanse
global_address_ cleanse_portugal	Portugal_AddressCleanse
global_address_ cleanse_spain	Spain_AddressCleanse
global_address_ cleanse_uk	UK_AddressCleanse
global_address_ cleanse_usa	USA_AddressCleanse
global_address_ cleanse_suggest	Global_AddressCleanse **Note:** Set options under Suggestion List Options.
japan_plugin	Japan_AddressCleanse

Data Quality sample transform	Equivalent Data Services transform configuration
multi_country_ plugin	Global_AddressCleanse **Note:** Multi-country plugin is now the EMEA engine and the Global Address engine within the Global Address Cleanse transform.
usa_plugin	Global_AddressCleanse **Note:** USA plugin is now the USA engine within the Global Address Cleanse transform.

Table 7-12: Global Suggestion Lists transform family

Data Quality sample transform	Equivalent Data Services transform configuration
global_suggest_lists_ global	GlobalSuggestions
	UKSuggestions

Table 7-13: Match transform family (non-priority)

Data Quality sample transform	Equivalent Data Services transform configuration
match_address	Address_MatchBatch
match_address_japan	AddressJapan_MatchBatch
match_comsumr_ househld_res_fam_ind	ConsumerHouseholdResFamInd_MatchBatch
match_comsumr_ househld_res_ind	ConsumerHouseholdResInd_MatchBatch
match_corporate_ househld	CorporateHouseholdFirmInd_MatchBatch
match_firm_address	FirmAddress_MatchBatch
match_firm_address_ japan	FirmAddressJapan_MatchBatch
match_individual_id	IndividualId_MatchBatch
match_name_address	NameAddress_MatchBatch

Data Quality sample transform	Equivalent Data Services transform configuration
match_name_date	NameDate_MatchBatch
match_name_email	NameEmail_MatchBatch
match_name_firm_ address	NameFirmAddress_MatchBatch
match_name_firm_ address_japan	NameFirmAddressJapan_MatchBatch
match_name_ individual_id	NameIndividualId_MatchBatch
match_name_phone	NamePhone_MatchBatch
match_ product_desc	ProductDescription_MatchBatch

Table 7-14: USA Regulatory Address Cleanse transform family

Data Quality sample transform	Equivalent Data Services transform configuration
usa_reg_address_ cleanse_CASS	USARegulatory_AddressCleanse
usa_reg_address_ cleanse_cass_ews	USARegulatoryEWS_AddressCleanse
usa_reg_address_ cleanse_cass_geo	USARegulatoryGeo_AddressCleanse
usa_reg_address_ cleanse_cass_rdi	USARegulatoryRDI_AddressCleanse
usa_reg_address_ cleanse_cass_z4c	USARegulatoryZ4Change_AddressCleanse
usa_reg_address_ cleanse_suggest	USARegulatorySuggestions_AddressCleanse

7.4.2 Address cleansing transforms

7.4.2.1 How Country ID transforms migrate

The `dqmigration` utility migrates the Data Quality Country ID transform directly to Country ID transform in Data Services.

Best Practice

You may notice that the Input, Options, and Output tabs contain options at the top of the tab to filter the fields listed. Filter options are **Best Practice**, **In Use**, and **All**. For migrated transforms, any fields or options that were set in the original transform are listed when you select either **Best practice** or **In use**.

Related Topics

- Overview of migrated transforms
- Migrating Data Quality output fields

7.4.2.2 How Global Address Cleanse transforms migrate

The `dqmigration` utility migrates Data Quality Global Address Cleanse transforms to a Global Address Cleanse transform in Data Services.

Most elements of the transform migrate to Data Services without any changes. For example, all of the options in the Standardization group migrate to the Global Address Cleanse transform in Data Services.

However, there are some aspects of the Data Quality Global Address Cleanse transform that change after migration to Data Services.

Note:

Global Address Cleanse migration error

In certain circumstances, when you migrate a Global Address Cleanse from Data Quality to Data Services, a validation error appears regarding the transforms output schema.

[EIM transform Source:<*name of Data Quality transform*>] BODI-1111071: Output column <*name of output column*> is not mapped.

When this error appears, open the transform and remap the field named in the error.

Related Topics

- Migration of Global Address Cleanse transform options

7.4.2.3 Migration of Global Address Cleanse transform options

When the `dqmigration` utility migrates your Global Address Cleanse transforms to Data Services, there are changes to certain options and option groups. The table below describes the changes to expect for Global Address Cleanse.

Data Quality Global Address Cleanse option	Description	Data Services Global Address Cleanse option
Plugins	Defines the options for processing addresses from different countries.	Now called “engines”. You define engine settings in the Options tab.
Multicountry plugin	Assigns the correct postal code for addresses from many different countries.	Replaced with the EMEA engine.
Field attributes	Defines the type of data a field contains as well as other aspects of the field.	Column headings in the Output tab of the transform that includes field name, field class, field category, type, and content type.
Suggestion lists	Settings that define the type of suggestion list information that is generated.	Settings that indicate whether or not to include certain suggestion list information.
Country Map	Option group that maps a country and script to a plugin for processing.	Deprecated and the migrated settings are mapped to the country and script code so that the best engine available processes the data.
Common option group	Option group that contains settings for transform performance.	Contains only one option; **Run as Separate Process**.
Component_Output_Fields, Standardized_Output_Fields, and Suggestion_Output_Fields	Option groups that define output field attributes.	Field attribute named Generated_Field_Category for output fields. Categories include Component, Standardized, and Suggestion.
Data_Record_Field_Type and Data_Record_Field_Length	Attributes for the output fields.	Field attributes that are listed in each transforms' Schema Out Type column.
Content_Type	Attributes for the output fields.	Field attribute that is listed in the Schema Out section in the Content Type column.

Data Quality Global Address Cleanse option	Description	Data Services Global Address Cleanse option
Remove Primary Name Apostrophe	Specifies whether to include punctuation in certain street names that include a DE L' or D'.	Attribute name changed to Remove Address Apostrophes.
Building_Name	A generated output field.	Output field name changed to Building_Name1.
Engines	Assigns the plugins to use for processing.	Deprecated and options migrated to respective engine group settings.

Best Practice

You may notice that the Input, Options, and Output tabs contain options at the top of the tab to filter the fields listed. Filter options are **Best Practice**, **In Use**, and **All**. For migrated projects, any fields or options that were set in the original transforms are listed when you select either **Best practice** or **In use**.

Related Topics

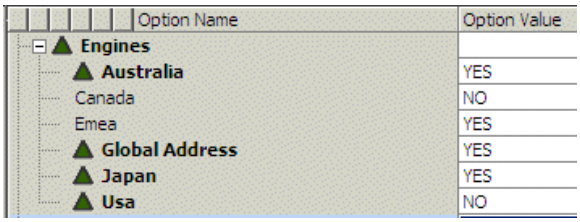
- Example of Global Address Cleanse migration
- Reference Guide: Transforms, Content types

7.4.2.4 Example of Global Address Cleanse migration

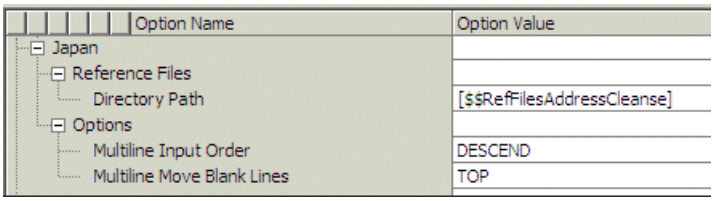
To give you a better idea of the changes in the Global Address Cleanse transform, suppose you have a Data Quality data flow that includes a Reader transform, a Global Address Cleanse France transform, and a Writer transform. Also included in the setup are plugins for Australia, Multi country, and Japan. The data flow in Data Quality looks like the following illustration:

The `dqmigration` utility creates a Data Services data flow that will not include plugins in the data flow. The `dqmigration` utility migrates the settings in the Data Services plugins to settings in the Global Address Cleanse France transform.

Also, the multi-country plugin settings are migrated to the EMEA engine setup.



The rest of the option settings in the Data Quality plugins migrate to the Global Address Cleanse transform into the applicable country sections. For example, the illustration below shows the Japan options.



7.4.2.5 How USA Regulatory Address Cleanse transforms migrate

The `dqmigration` utility migrates Data Quality USA Regulatory Address Cleanse transform to the Data Services USA Regulatory Address Cleanse transform.

Note:

- If you used DPV or LACSLink in Data Quality, and did not purchase these options in Data Services, you must disable these options before migrating. If you do not disable them, your job will not run in Data Services.
- The `dqmigration` utility migrates the GeoCensus functionality of the Data Quality USA Regulatory Address Cleanse transform to the Data Services USA Regulatory Address Cleanse transform. However, the GeoCensus functionality in the Data Services USA Regulatory Address Cleanse transform will be deprecated in a future version. It is recommended that you upgrade any data flows that currently use the GeoCensus functionality to use the Geocoder transform. For instructions on manually upgrading from GeoCensus to the Geocoder transform, see the *Data Services Behavior Changes* section.

Related Topics

- Migrating Data Quality output fields
- Migrating USA Regulatory Address Cleanse Options
- GeoCensus and Geocoder

7.4.2.6 Migrating Data Quality output fields

After you migrate the Data Quality USA Regulatory Address Cleanse transform to Data Services using the `dqmigration` utility, you will notice changes in how output fields are listed, mapped, and defined.

Best Practice

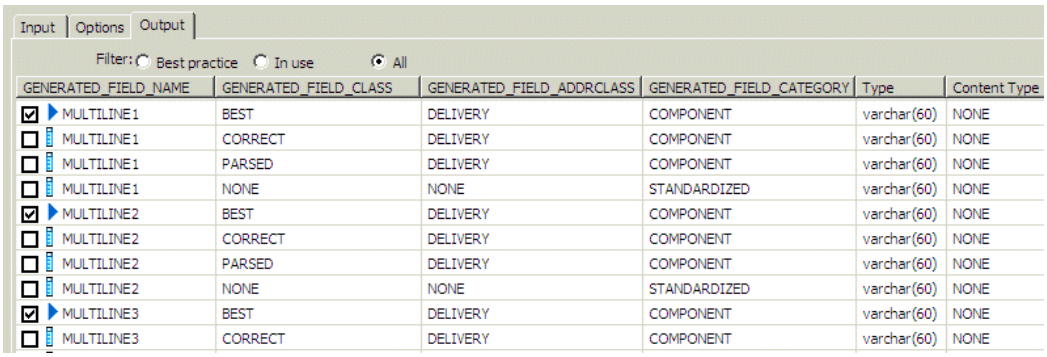
You may notice that the Input, Options, and Output tabs contain options at the top of the tab to filter the fields listed. Filter options are **Best Practice**, **In Use**, and **All**. For migrated projects, any fields or options that were set in the original project are listed when you select either **Best practice** or **In use**.

Output field types

All output fields, Component, Standardized, Suggestion, and Suggestion List, are migrated to the Output tab in Data Services. The Output tab lists the output fields and their attributes, which are populated with the settings in Data Quality. Attributes are listed under the following column headings: Generated_Field_Name, Generated_Field_Class, Generated_Field_AddrClass, Generated_Field_Category, Type, and Content Type.

Example:

Data ServicesUSA Regulatory Address Cleanse


Input | Options | Output

Filter: Best practice In use All

GENERATED_FIELD_NAME	GENERATED_FIELD_CLASS	GENERATED_FIELD_ADDRCLASS	GENERATED_FIELD_CATEGORY	Type	Content Type
MULTILINE1	BEST	DELIVERY	COMPONENT	varchar(60)	NONE
MULTILINE1	CORRECT	DELIVERY	COMPONENT	varchar(60)	NONE
MULTILINE1	PARSED	DELIVERY	COMPONENT	varchar(60)	NONE
MULTILINE1	NONE	NONE	STANDARDIZED	varchar(60)	NONE
MULTILINE2	BEST	DELIVERY	COMPONENT	varchar(60)	NONE
MULTILINE2	CORRECT	DELIVERY	COMPONENT	varchar(60)	NONE
MULTILINE2	PARSED	DELIVERY	COMPONENT	varchar(60)	NONE
MULTILINE2	NONE	NONE	STANDARDIZED	varchar(60)	NONE
MULTILINE3	BEST	DELIVERY	COMPONENT	varchar(60)	NONE
MULTILINE3	CORRECT	DELIVERY	COMPONENT	varchar(60)	NONE

Note:

You can change attributes such as field length by double-clicking the name of the field in the Output Schema portion of the transform. For more information about changing field attributes, see the *Administrator's Guide*.

Output_Fields/Field

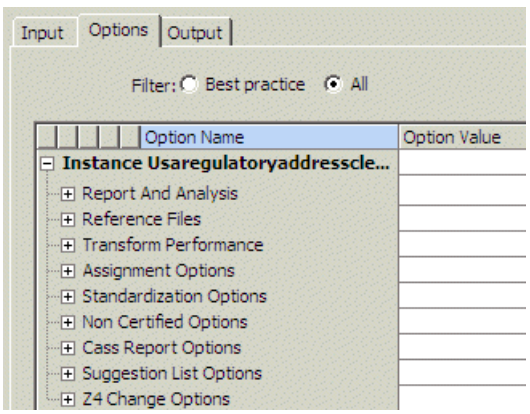
The following fields migrate to the Schema Out section of Data Services.

Data Quality transform option	Description	Data Services option
Content_Type	Specifies the type of data in the field.	Schema Out Content Type.
Data_Record_ Field_Length	Specifies the number of characters to make available for the posted data in this field.	Schema Out Type
Data_Record_ Field_Type	Specifies the kind of data this output field will contain.	Schema Out Type

Transform options: All transform options (Assignment, Standardization, and so on) appear under the Options tab in Data Services.

Example:

Data Services USA Regulatory Address Cleanse



Related Topics

- Reference Guide: Content types

7.4.2.7 Migrating USA Regulatory Address Cleanse Options

The following table shows how the `dqmigration` utility maps the Data Quality USA Regulatory Address Cleanse options to Data Services.

Data Quality USA Regulatory Address Cleanse option	Description	Data Services USA Regulatory Address Cleanse option
Option Group settings (values)	Options that you set in the transform to specify how you want the transform to process your data	The values are migrated to the Options tab. Each category of options make up a group of options. For example, the assignment options are in a group labeled Assignment Options.
Suggestion_Lists_Output_Fields	Options that you set in the transform to specify what information to include in suggestion list output.	The values are migrated to the Options tab under the Suggestion List Options group. Each group is divided into Lastline Components, Primary Address Components, and Secondary Address Components.
Component_Output_Fields, Standardized_Output_Fields, and Suggestion_Output_Fields	In Data Quality, there are option groups for Component, Standardized, and Suggestion output fields.	In Data Services, output fields are listed in the Output tab, and they are assigned a Generated_Field_Category of Component, Standardized, or Suggestion.
Data_Record_Field_Type, Content_Type, and Data_Record_Field_Length	In Data Quality, these three attributes are set in the Output Fields groups.	In Data Services, these fields don't exist. Instead, the settings are migrated into the Type field attribute, which is a column in the Output tab.

Best Practice

You may notice that the Input, Options, and Output tabs contain options at the top of the tab to filter the fields listed. Filter options are **Best Practice**, **In Use**, and **All**. For migrated projects, any fields or options that were set in the original project are listed when you select either **Best practice** or **In use**.

Pre-migration task for jobs with LACSLink and DPV

Read this section only if you have purchased the LACSLink and DPV features in Data Quality, but you did not purchase these features in Data Services.

If you migrate a Data Quality job to Data Services that has LACSLink and DPV enabled, but you haven't purchased the LACSLink and DPV features in Data Services, you may need to build the job in Data Services manually.

For example, suppose you have purchased LACSLink and DPV for Data Quality XI, and you have jobs that are set up for LACSLink and DPV processing. Then suppose you upgrade to Data Services, but

you do not have LACSLink and DPV features. The `dqmigration` utility migrates the jobs successfully. However, when you import the jobs into Data ServicesDesigner, the LACSLink and DPV features become disabled.

In addition, if you open the job that contains the LACSLink and DPV transform configuration, the transform corresponding to that configuration on the canvas will have a red X on the corner indicating that it is broken. If this occurs, you must delete the job and create a new one.

To avoid creating a new job, disable LACSLink and DPV in Data Quality before you migrate to Data Services. Then the `dqmigration` utility migrates the job successfully, and you can import the job to Data ServicesDesigner without ruining the job.

If you have migrated and opened your job, and the transform is disabled (has a red X on the node in the canvas), you can purchase the LACSLink or DPV features (and activate them with the assigned keycode) and then use the jobs.

Related Topics

- Migrating Data Quality output fields

7.4.2.8 Post-migration tasks for USA Regulatory Address Cleanse GeoCensus

The `dqmigration` utility migrates the GeoCensus functionality of the Data Quality USA Regulatory Address Cleanse transform to the Data Services USA Regulatory Address Cleanse transform. However, the GeoCensus functionality in the Data Services USA Regulatory Address Cleanse transform will be deprecated in a future version. It is recommended that you upgrade any data flows that currently use the GeoCensus functionality to use the Geocoder transform. For instructions on manually upgrading from GeoCensus to the Geocoder transform, see the *Data Services Behavior Changes* section.

Related Topics

- GeoCensus and Geocoder

7.4.3 Reader and Writer transforms

7.4.3.1 How database Reader transforms migrate

The `dqmigration` utility migrates a Reader transform with a database data source to the following Data Services objects:

- SQL transform -- Performs the indicated SQL query operation.
- Query transform -- Maps the Reader fields to the Data Quality input fields.

Note:
After migration, open the resulting SQL transform in the migrated data flow and click Update Schema to sequence the columns in the order of the SQL query, and open each downstream transform to update the columns to correspond to the updated SQL transform schema.

This section covers the following topics:

- Mapping database Reader transform options
- Example of migrated database Reader
- Database Reader with Pre and Post SQL Statements

7.4.3.1.1 Mapping database Reader transform options

The following tables show how database Reader options map to the Data Services object and options.

Table 7-18: Transform_Performance option

Data Quality database Reader option	Description	Data Services object and option	Description of Data Services option
Bulk_Read_ Size_Recs	Specifies the bulk factor and controls whether or not bulk reading is used. This option is for relational databases only.	SQL transform **Array fetch size**	Indicates the number of rows retrieved in a single request to a source database.

Table 7-19: Connection_Options

Data Quality database Reader option	Description	Data Services object and option
Data_Source_ Name	Specifies the name and path to the data source.	Datastore **Data source**
Driver_Name	Specifies the database driver to use when connecting from the Reader to the data source.	See table in How connections migrate
Host_Name	Specifies the server name where the data source resides. This option is for relational databases only.	Ignored

Data Quality database Reader option	Description	Data Services object and option
Named_ Connection	Specifies the name of a pre-configured data source connection to either a relational database or a flat file.	Becomes part of the name of the datastore.
Password	Specifies the password appropriate to the user name in order to access the relational database.	Datastore **Password**
Port_Number	Specifies the port number used for connections. This option is for relational databases only.	Ignored
User_Name	Specifies the login name required to access the relational database.	Datastore **User name**

Table 7-20: Misc_Options

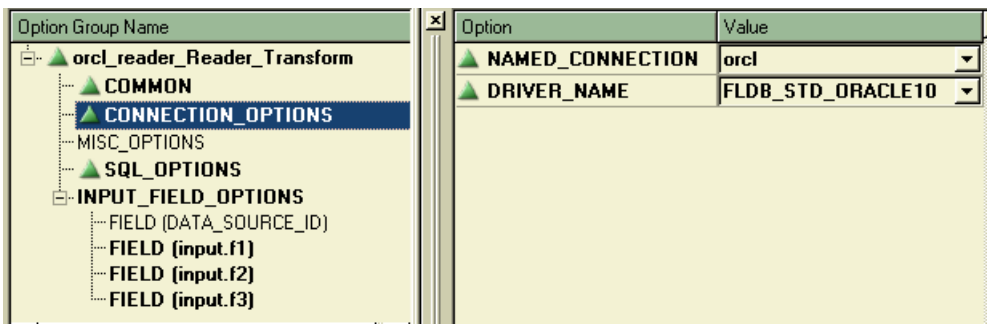
Data Quality database Reader option	Description	Data Services object and option
Data_Source_ID	Adds a new column and value that you specify before it passes to the next transform.	Query transform Schema Out column DATA_SOURCE_ID, which contains the constant value defined in this option in the Data Quality Reader.
Max_Records_Per_ Collection	Specifies the number of records allowed in each collection before it is sent to the next transform.	Ignored
Max_Records_ Processed	Specifies the total number of records used for processing.	Ignored
Starting_Record_ Num	Specifies the record number that you want to include first. This option is for flat files only.	Ignored
Unique_Record_ID_ Stem	Adds a new column and value that you specify used for sequencing.	Query transform Schema Out column UNIQUE_RECORD_ID_STEM which contains the constant value defined in this option in the Data Quality Reader.

Table 7-21: SQL_Options

Data Quality database Reader option	Description	Data Services object and option	Description of Data Services option
Post_SQL_Error_ Action	Sets the action that you want to occur if a Post_SQL_Statement has failed.	Ignored	Not applicable
Post_SQL_ Statement	Specifies the SQL to execute after the Select_SQL_Statement.	Post-Dataflow Script that contains SQL function	SQL function runs a SQL operation against tables in the specified database
Pre_SQL_Error_ Action	Sets the action that you want to occur if a Pre_SQL_Statement has failed.	Ignored	Not applicable
Pre_SQL_ Statement	Specifies the SQL to execute before the Select_SQL_Statement begins.	Pre-dataflow script that contains SQL function	SQL function runs a SQL operation against tables in the specified database
Select_SQL_ Statement	Specifies the records that are selected from the input source, which has a driver that accepts SQL commands.	SQL Transform **SQL Text**	Text of the SQL query that is passed to the database server

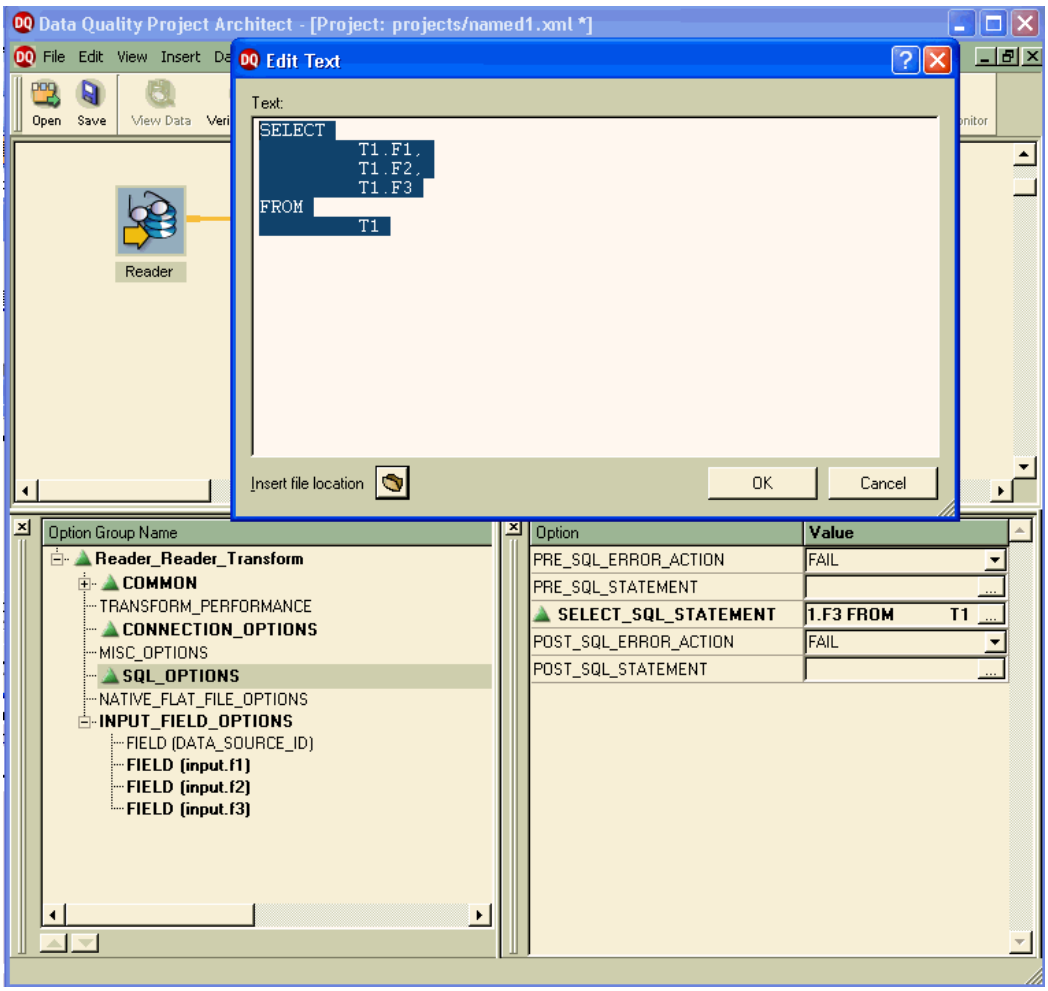
7.4.3.1.2 Example of migrated database Reader

This example shows how the `dqmigration` utility migrates a Data Quality project with a database Reader.

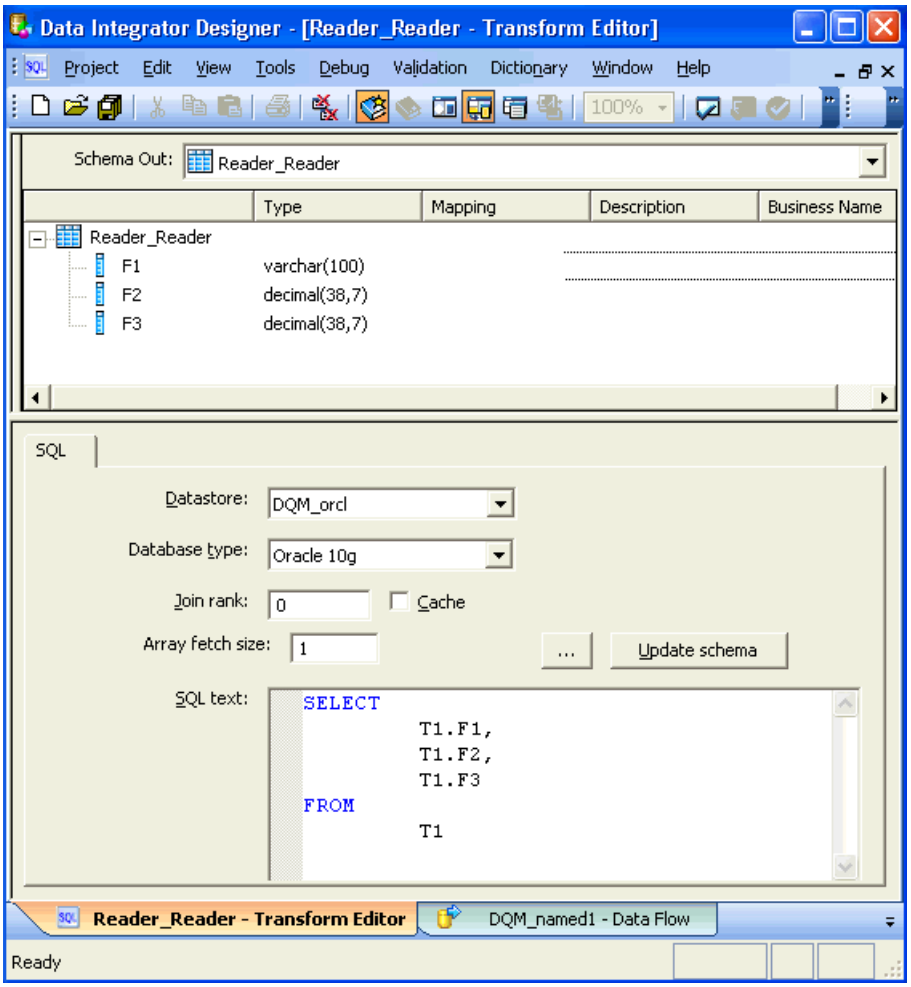


This database Reader transform uses a Named_Connection `orcl`. The `dqmigration` utility migrates the Named_Connection to the datastore `DQM_orcl`. For details about how the named connection options map to the datastore options, see How connections migrate.

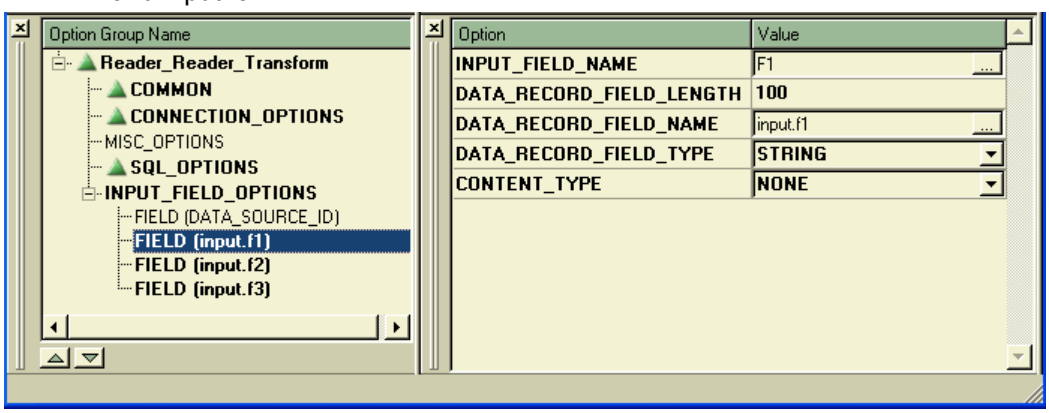
This sample Reader has the following SQL_Options group. Option Select_SQL_Statement selects the input database columns F1, F2, and F3.



The `dqmigration` utility migrates this SELECT statement to the Data Services SQL transform. In the following SQL Transform Editor, the SQL transform named Reader_Reader contains this SELECT statement in the SQL Text option.

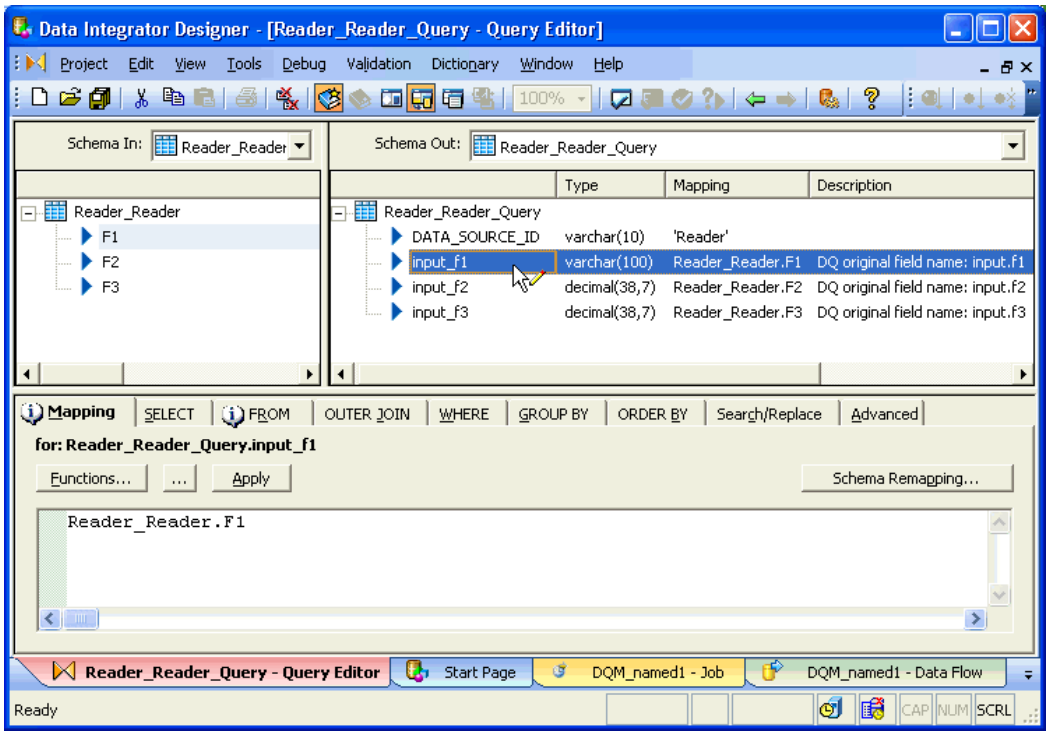


The Data Quality Reader maps the database columns f1 , f2, and f3 to input fields input.f1, input.f2 , and input.f3.



The `dqmigration` utility migrates this mapping to a Data Services query transform.

However, Data Services does not allow a period (.) in the column names. Therefore, the `dqmigration` utility replaces the period with an underscore (_). The sample Data Services query transform named Reader_Reader_Query maps the input field names `F1`, `F2`, and `F3` to column names `input_f1`, `input_f2`, and `input_f3`.



The following table shows how the data types from the sample Data Quality Reader input fields map to the data types in Data Services.

Table 7-22: Mapping of input data types

Data Quality database Reader input field option	Data Quality Value	Data Services Query transform option	Data Services Value
Input_Field_Name	F1	Schema Out column name	input_F1
Data_Record_Field_Name	input.f1		
Data_Record_Field_Length	100	Column data type	varchar(100)
Data_Record_Field_Type	String		

Input_Field_Name	F2	Schema Out column name	input_F2
Data_Record_Field_Name	input.f2		
Data_Record_Field_Length	38	Column data type	decimal(38,7)
Data_Record_Field_Type	Number		

Related Topics

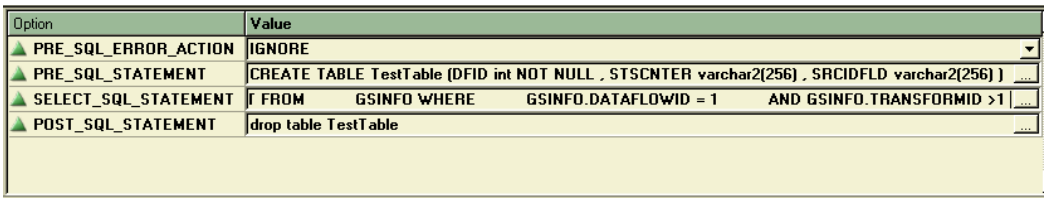
- How connections migrate
- How data types migrate

7.4.3.1.3 Database Reader with Pre and Post SQL Statements

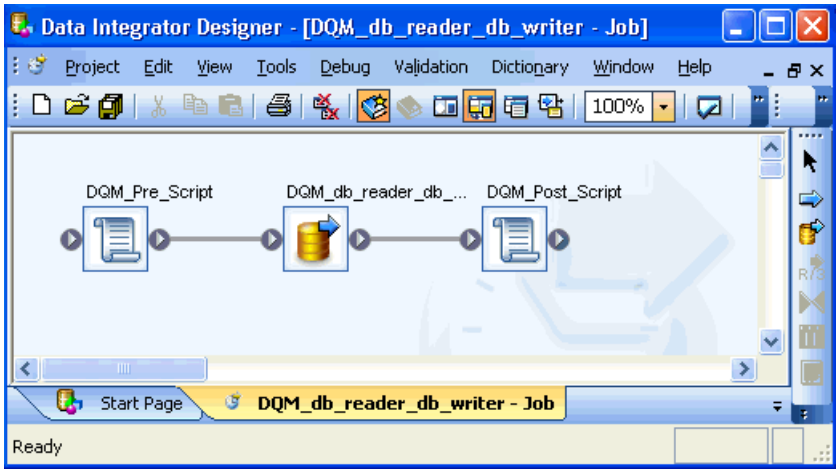
If your Data Quality transform contains pre or post SQL statements, the `dqmigration` utility migrates them to pre or post scripts around the Data Services data flow.

Example: How a database reader with pre and post SQL statements migrate

Suppose your Data Quality project db_reader_db_writer has a Pre_SQL_Statement option that creates a test table and a Post_SQL_Statement that drops the table.



The resulting migrated job DQM_db_reader_db_writer contains a DQM_Pre_Script and DQM_Post_Script around the data flow DQM_db_reader_db_wrtier.



The resulting migrated DQM_Pre_Script contains the Pre_SQL_Statement value:

```
sql('DQM_db_reader_db_writer_My_DB_Reader_Reader',
'CREATE TABLE TestTable (DFID int NOT NULL ,
    STSCNTER varchar2(256) ,
    SRCIDFLD varchar2(256) )');
```

The resulting migrated DQM_Post_Script contains the Post_SQL_Statement value :

```
sql('DQM_db_reader_db_writer_My_DB_Reader_Reader',
 'drop table TestTable');
```

7.4.3.2 How database Writer transforms migrate

The `dqmigration` utility migrates a database Writer transform to the following Data Services objects:

- Query transform -- Maps the fields from the previous transform to the Target columns.

  The generated name for this query has the format:

  ```
  DataQualityWriterName_Writer_Query
  ```

- Target -- Contains the SQL statements from the Data Quality Writer transform options **Write_SQL_Statement**, **Pre_Write_SQL_Statement**, and **Post_Write_SQL_Statement**.

  A target is an object to which Data Services loads extracted and transformed data in a data flow. However, the target table that results from a migrated Data Quality Writer is not the object that is loaded. The SQL statement in Write_SQL_Statement identifies the table into which you insert, update, or delete, and this table might be created by the **Pre_Write_SQL_Statement**. When you execute the migrated job, Data Services executes these SQL statements.

The generated name for this target has the format:

```
DataQualityWriterName_Writer
```

For example, if your Data Quality Writer name is `My_DB_Writer`, it becomes the target named `My_DB_Writer_Writer` after migration to Data Services.

- Datastore -- Fulfills the target object requirement for a table defined in a datastore.

  A Data Services Target object requires a table and schema defined in a datastore. The `dqmigration` utility creates a dummy table and datastore and places their names in the target object. This target table does not appear in the imported table list of the datastore. You can ignore this generated datastore unless you have defined a similarly named table that is accessed by SQL statements elsewhere and you want to change the name for clarity.

The following topics describe the mapping of the Data Quality database Writer options to the Data Services Query and Target options.

- Mapping database Writer Connection_Options
- Mapping Per_Write_SQL_Options
- Mapping Per_Write_SQL_Options
- Mapping Output_Field_Options

7.4.3.2.1 Mapping database Writer Connection_Options

The Connection_Options option group includes information about accessing the output destination, which can be a flat file or a relational database. The following table shows how the `dqmigration` utility maps these Data Quality options to Data Services datastore options.

Table 7-23: Mapping Connection_Options

Data Quality database Writer option	Description of Data Quality option	Data Services object and option	Description of Data Services option
Data_Source_Name	Specifies the name and path to the data source.	Datastore **Data source** for DB2, MSSQL, ODBC Datastore **Connection name** for Oracle	"Datastores" represent connection configurations between Data Services and databases or applications.
Driver_Name	Specifies the database driver to use when connecting from the Writer to the output destination.	See table in How connections migrate	Database type and database version

Data Quality database Writer option	Description of Data Quality option	Data Services object and option	Description of Data Services option
Host_Name	Specifies the server name where the output destination resides. This option is for relational databases only.	Ignored, but the `dqmigration` log displays a warning.	Not applicable
Named_ Connection	Specifies the name of a preconfigured data source connection to either a relational database or a flat file.	Datastore **Connection name** File Format **Name**	Descriptive name for the database connection Descriptive name for the file format
Password	Specifies the password appropriate to the user name in order to access the relational database for output.	Datastore **Password**	User's password
Port_Number	Specifies the port number used for connections. This option is for relational databases only.	Ignored, but the `dqmigration` log displays a warning.	Not applicable
User_Name	Specifies the login name required to access the relational database for output.	Datastore **User name**	User name of the account through which Data Services accesses the database.

7.4.3.2.2 Mapping Once_Per_Writer_SQL_Options

The SQL statements in the Once_Per_Writer_SQL_Options are executed only once per Writer transform.

Note:
Migrated jobs ignore all *_Error_Action options. If an SQL statement fails, the Data Services job fails.

Table 7-24: Once_Per_Writer_SQL_Options

Data Quality database Writer option	Description	Data Services object and option
Post_Connection_ Error_Action	Sets the action that you want to occur if a Post_Connection_SQL_Statement has failed.	Ignored

Data Quality database Writer option	Description	Data Services object and option
Post_Connection_ SQL_Statement	Specifies the SQL statement to execute after a connection to the database is established.	In Target object, on Pre-Load Commands tab, first part of **SQL Commands** text box
Post_SQL_ Error_Action	Sets the action that you want to occur if a Post_SQL_Statement has failed.	Ignored
Post_SQL_ Statement	Specifies the SQL statement to execute after the Select_SQL_Statement.	In Target object, on Post-Load Commands tab, part of **SQL Commands** text box
Pre_SQL_ Error_Action	Sets the action that you want to occur if a Pre_SQL_Statement has failed.	Ignored
Pre_SQL_ Statement	Specifies the SQL to execute before the Select_SQL_Statement begins.	In Target object, on Pre-Load Commands tab, first part of **SQL Commands**

For example, suppose your Writer transform contains the following values for the Once_Per_Writer_SQL_Options.

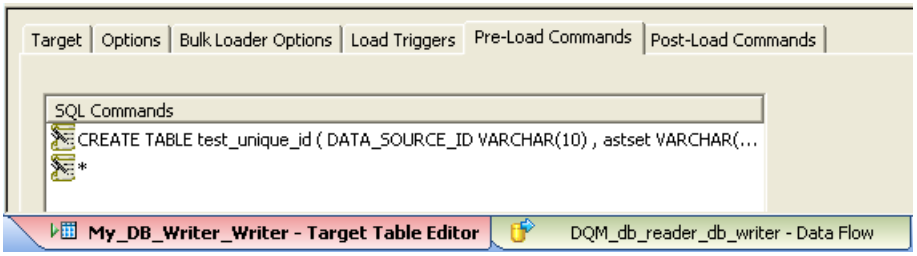
- **Pre_SQL_Statement** option:

```
CREATE TABLE test_unique_id
(
DATA_SOURCE_ID         VARCHAR(10)   ,
astset                 VARCHAR(256)  ,
dataflowid             NUMERIC(10)   ,
mtlvlname              VARCHAR(256)  ,
mtset                  VARCHAR(256)  ,
transformid            NUMERIC(10)   ,
UNIQUE_RECORD_ID_STEM  VARCHAR(10)
);
```

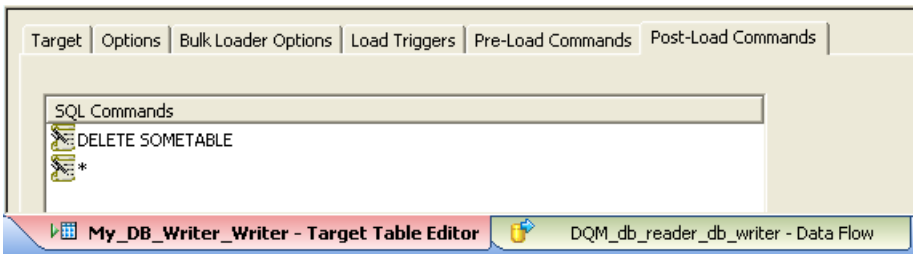
- **Post_SQL_Statement** option:

```
DELETE SOMETABLE
```

The `dqmigration` utility produces the following dummy Target object with the Pre-Load Commands tab that contains the **Pre_SQL_Statement** value.



The following Post-Load Commands tab shows the **Post_SQL_Statement** value.



7.4.3.2.3 Mapping Per_Write_SQL_Options

In Data Quality, any SQL statements entered in Per_Write_SQL_Options options are processed for each data collection. In Data Services, data collections do not exist and these SQL statements are processed for the entire data set.

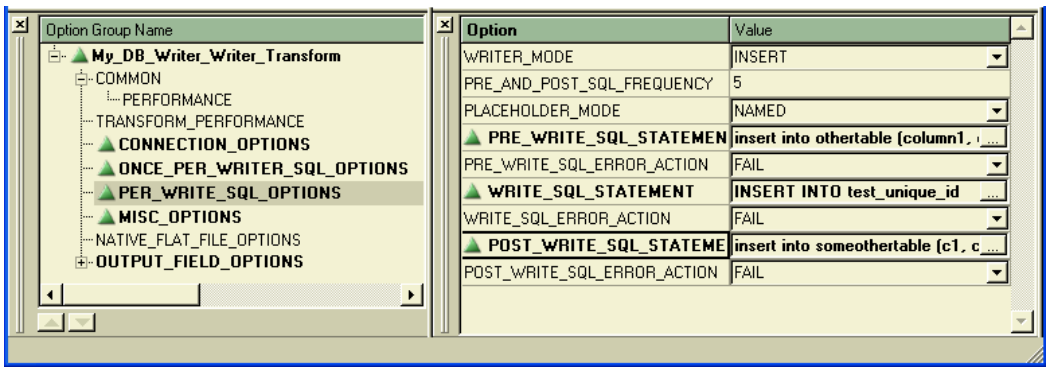
Note:
Migrated jobs ignore all *_Error_Action options. If an SQL statement fails, the Data Services job fails.

Table 7-25: Per_Write_SQL_Options

Data Quality database Writer option	Description	Data Services object and option
Placeholder_Mode	Specifies whether you want placeholders in your SQL statements, and if so, the type of placeholder.	In Target object, on Options tab, option **Column comparison** is set to `Compare by name`
Post_Write_SQL_Error_Action	Sets the action that you want to occur if a Post_Write_SQL_Statement has an error.	Ignored
Post_Write_SQL_Statement	Specifies the SQL to execute after the file is written.	In Target object, on Load Triggers tab, third SQL statement
Pre_And_Post_SQL_Frequency	Specifies how often the Pre_Write_SQL_Statement and Post_Write_SQL_Statement should be invoked.	Ignored

Data Quality database Writer option	Description	Data Services object and option
Pre_Write_SQL_ Error_Action	Sets the action that you want to occur if a Pre_Write_SQL_Statement has an error.	Ignored
Pre_Write_SQL_ Statement	Specifies database operations before writing a record.	In Target object, on Load Triggers tab, first SQL statement
Write_SQL_Error_ Action	Sets the action that you want to occur if a Write_SQL_Statement has an error.	Ignored
Write_SQL_ Statement	Specifies the records that are placed into the output destination, which has a driver that accepts SQL commands.	In Target object, on Load Triggers tab, second SQL statement
Writer_Mode	Sets a mode for performing SQL commands.	In Target object, on Load Triggers tab, option **On Operation**

For example, suppose your Writer transform contains the following values for the Per_Write_SQL_Options to insert into the `test_unique_id` table (created by the Pre_SQL_Statement) with Placeholder_Mode value `Named`.



Option Pre_Write_SQL_Statement contains the following SQL statement:

```
insert into othertable (column1, column2) values (:ds, :at)
```

Option Write_SQL_Statement contains an insert into the `test_unique_id` table:

```
INSERT INTO test_unique_id
    (DATA_SOURCE_ID,
     astset,
     dataflowid,
     mtlvlname,
     mtset,
     transformid,
UNIQUE_RECORD_ID_STEM)
VALUES
     (:ds,
      :at,
```

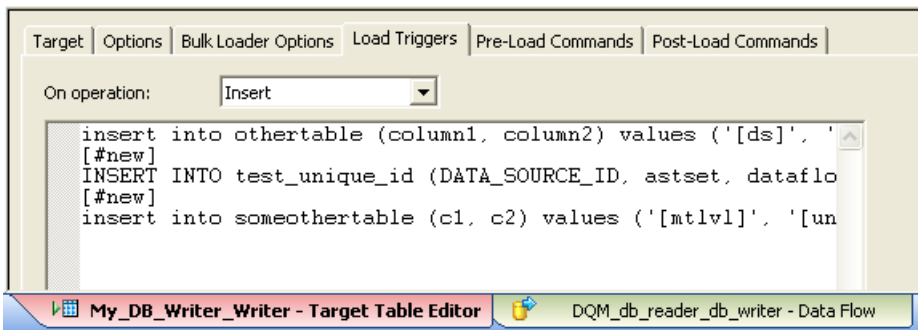
```
      :df,
      :mtlvl,
      :mt,
      :tf,
      :unq);
```

Option Post_Write_SQL_Statement contains the following SQL statement:

```
insert into someothertable (c1, c2) values (:mtlvl, :unq)
```

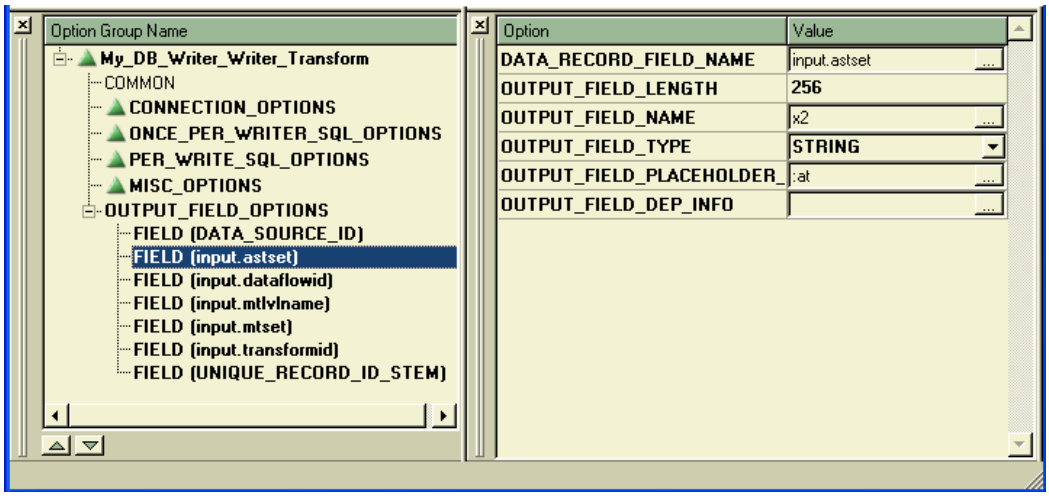
The `dqmigration` utility produces the following Target object with the Load_Triggers tab that contains these three SQL statements:

- The **Pre_Write_SQL_Statement** value as the first statement
- The **Write_SQL_Statement** value as the second statement
- The **Post_Write_SQL_Statement** value as the third statement



7.4.3.2.4 Mapping Output_Field_Options

The Output_Field_Options option group creates the output fields that hold the data that this transform creates. For example, the following Output_Field_Options show the input field name in each Field option. The Placeholder_Mode option in the Per_Write_SQL_Options option group has a value `Named` (which is the default value) that indicates the Writer should map each input field name to the Output_Field_Place holder_Name.



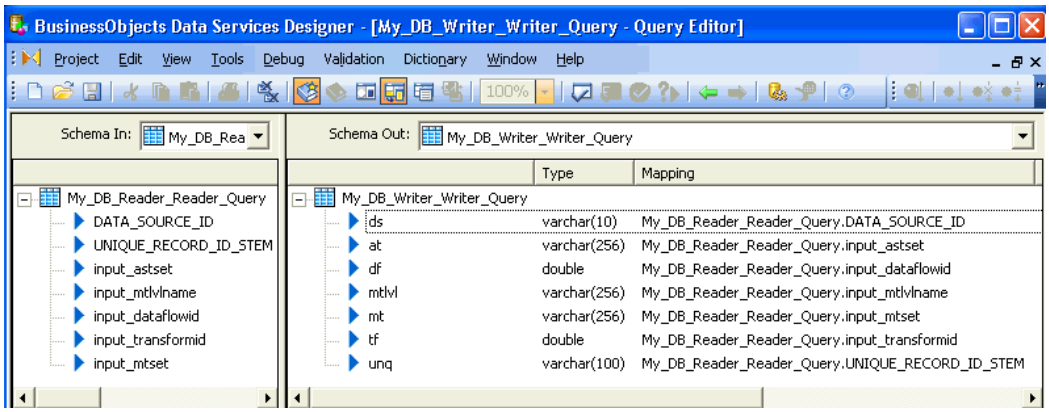
After the `dqmigration` utility migrates this Writer transform, the Data Services Query transform maps the input field names to the placeholder names. However, a colon (:) is not allowed in Data Services names. Therefore, the `dqmigration` utility removes the colon.

Table 7-26: Mapping of input field names to output placeholder names

Data Quality Output_Field_Options Field	Data Quality Output_Field_ Placeholder name	Data Services column name
Data_Source_ID	:ds	ds
input.astset	:at	at
input.dataflowid	:df	df
input.mtlvlname	:mtlvl	mtlvl
input.mtset	:mt	mt
input.transformid	:tf	tf
Unique_Record_ID_Stem	:unq	unq

The `dqmigration` utility produces the following Query transform named My_DB_Writer_Writer_Query which shows:

- The input field names are in the Schema In area of the Query transform.
- The modified placeholder names are in the Schema Out area.
- The Mapping column in Schema Out contains the name of the input field, prefaced with the name of the object before this Query.



7.4.3.2.5 Ignored database Writer options

The `dqmigration` utility ignores the following database Writer option groups.

Table 7-27: Transform_Performance, Misc_Options, and Field option groups

Option group	Data Quality database Writer option	Result of dqmigration utility	Comments
Transform_ Performance	Bulk_Insert_Size_ Recs	Ignored	Create a new target table and datastore and replace the migrated target table to take advantage of the Data Services bulkloading feature.
Misc_Options	Max_Records_Processed	Ignored	Data Services provides a pageable cache which allows a data flow to process more than two gigabytes of data.
Field	Output_Field_ Dep_Info	Ignored	

Related Topics

- Fixing invalid datastores

7.4.3.3 How flat file Reader and Writer transforms migrate

7.4.3.3.1 How flat file Reader transforms migrate

The `dqmigration` utility migrates Reader transforms with flat file inputs to the following Data Services objects:

- File Format -- A file format is a set of properties describing the structure of a flat file (ASCII). File formats describe the metadata structure which can be fixed or delimited. You can use one file format to access multiple files if they all have the same metadata structure.
- Source -- A source is an object from which Data Services reads data. In this case, it is the specific flat file from which the migrated Data Quality Reader transform reads data.
- Query transform -- A Query transform maps the Reader fields to the Data Quality input fields.

The following topics describe the mapping of the Data Quality flat file Reader options to the Data Services Source and Query options.

- Migration of flat file Reader Connection_Options
- Mapping flat file Reader Misc_Options
- Mapping Native_Flat_File_Options
- Mapping Input_Field_Options
- Ignored flat file Reader options

Migration of flat file Reader Connection_Options

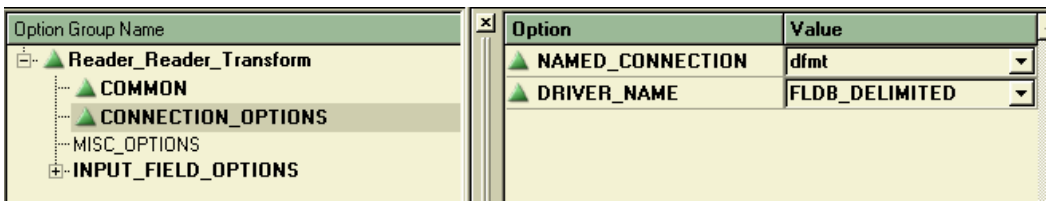
The Connection_Options group includes information about accessing the input source for a flat file Reader transform. The input source can be a flat file such as fixed-length ASCII or delimited.

The following table shows how the `dqmigration` utility maps the flat file Reader Connection_Options to the Data Service File Format options.

Data Quality database Reader option	Description	Data Services File Format section and option
Named_ Connection	Specifies the name of a pre-configured data source connection to either a relational database or a flat file.	**General** section, **Name** contains the connection name prefaced with `DQM`. **File Name(s)** contains the Data_Source_Name to which the named connection refers.

Data Quality database Reader option	Description	Data Services File Format section and option
Data_ Source_Name	Specifies the name and path to the data source.	**General** section, **Name** contains the data source name. **Data File(s)** section, **File Name(s)** contains the path.
	Contains a substitution variable for the name and path to the data source.	**General** section, **Name** contains the data source name. **Data File(s)** section, **File Name(s)** contains the substitution parameter. **Note:** The term “substitution variable” is changed to “substitution parameter” to avoid confusion with existing Data Services variables.
Driver_Name	Specifies the flat file driver to use when connecting from the Reader to the data source. For flat files, this value is either FLDB_DELIMITED or FLDB_FIXED_ASCII.	**General** section, **Type** contains `Delimited` or `Fixed Width`.

For example, suppose you have a Data Quality project `ff_named.xml` that has the following Connection_Options values. The named connection `dfmt` is for data source `e:\Titan\ff.txt`.



Using the information in the named_connections file, the `dqmigration` utility creates a Data Services file format with the following values for this Reader transform:

- **Name**: `DQM_Dfmt`
- **Type**: `Delimited`
- **File Name(s)**: `e:\Titan\ff.txt`

Mapping flat file Reader Misc_Options

The Misc_Options group includes various information about record size, unique records, and more.

The following table shows how the dqmigration utility maps the Data Quality Misc_Options to the Data Service Query transform and File Format options.

Data Quality flat file Reader option	Data Services object and option
Data_Source_ID	Query transform, Schema Out column DATA_SOURCE_ID which contains the constant value defined in this option in the Data Quality Reader.
Max_Records_Per_Collection	Ignored
Max_Records_ Processed	Ignored
Starting_Record_Num	A File Format that: • has option **Skipped rows** set to this Data Quality option value minus one. Therefore, this File Format starts with this record number. If another flat file Reader has a different value for Starting_Record_Num set, then migration creates a new File Format. • Has a name with "_2" appended to it. See example below.
Unique_Record_ID_Stem	Query transform, Schema Out column UNIQUE_RECORD_ID_STEM which contains the constant value defined in this option in the Data Quality Reader.

Example: How Starting_Record_Num migrates

Suppose you have a Data Quality project named ff_input.xml and the value of **Starting_Record_Num** is 1000. After migration:

- The new Data Services File Format has the name DQM_ff_input_2.
- The value of **Skipped rows** is 999.

Mapping Native_Flat_File_Options

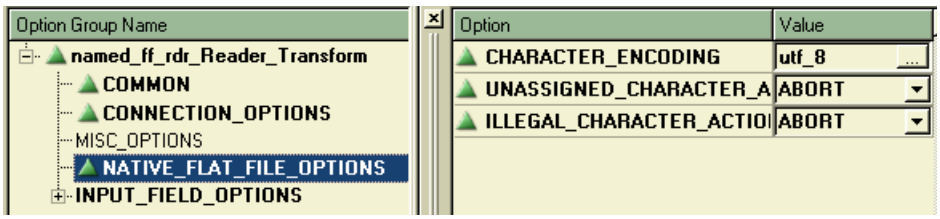
The Native_Flat_File_Options options group provides information about fixed-length ASCII or delimited files.

Note:
dBASE3 files are not supported in Data Services. The dqmigration utility migrates a dBASE3 file to a **Fixed width** file type in Data Services.

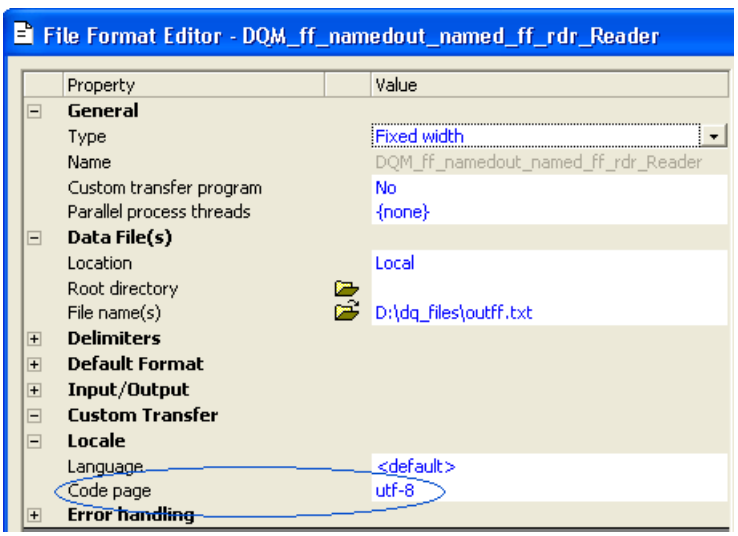
The following table shows how the dqmigration utility maps the Native_Flat_File_Options to the Data Service File Format options.

Data Quality flat file Reader option	Data Services object and option
Default_Flat_File_Schema_Location	The `dqmigration` utility looks in this location for the fmt or dmt file to define the File Format in Data Services.
Character_Encoding	File Format , Locale section **Code Page**
Unassigned_Character_Action	Ignored
Unassigned_Character_Default	Ignored
Illegal_Character_Action	Ignored
Illegal_Character_Default	Ignored
Unicode_To_ASCII_Table	Ignored

For example, suppose you have a Data Quality project `ff_namedout.xml` that has the following Native_Flat_File_Options values.



The following Data Services File Format Editor shows that the option **Character_Encoding** value `utf-8` migrated to the Data Services **Code Page** option and the other options are ignored.



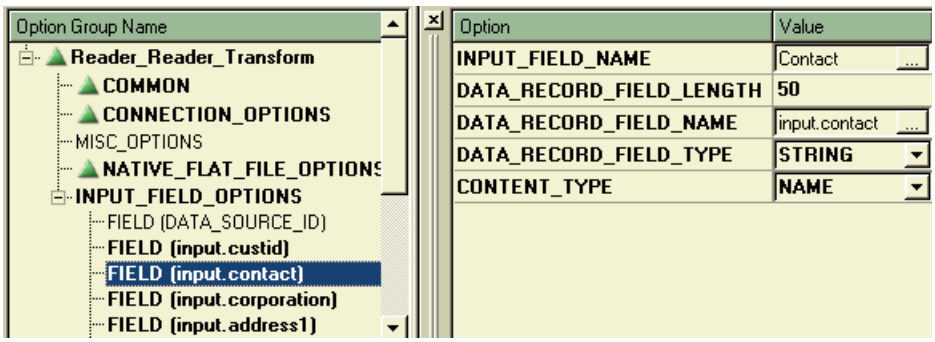
Mapping Input_Field_Options

The Input_Field_Options option group maps the input fields to use in this Reader transform.

The following table shows how the `dqmigration` utility maps the Input_Field_Options to the Data Service Query transform schema attributes.

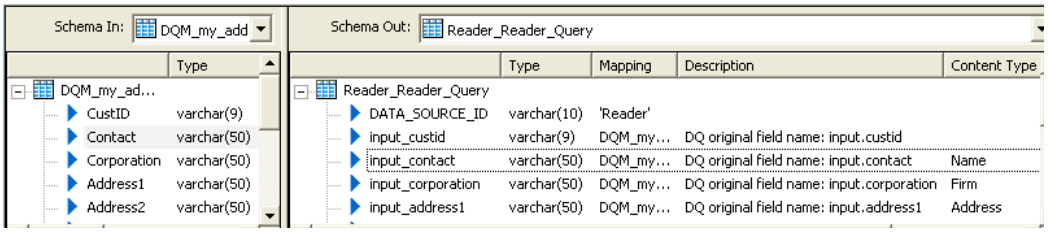
Data Quality Reader Field option	Data Services Query schema attribute
Content_Type	Schema Out **Content Type**
Data_Record_Field_Length	Schema Out **Type** contains length
Data_Record_Field_Name	Schema Out column name
Data_Record_Field_Type	Schema Out **Type**
Input_Field_Name	Schema In column name

For example, suppose you have a Data Quality project `my_address_cleanse_usa.xml` that has the following Input_Field_Options and options for Field (input.contact).



The following sample Data Services Query Editor shows:

- The Data Quality **Input_Field_Name** value `Contact` migrates to a Schema In column in the Data Services Query transform.
- The Data Quality **Data_Record_Field_Name** value `input.custid` migrates to a Schema Out column in the Data Services Query transform.
- The Data Quality **Data_Record_Field_Length** and **Data_Record_Field_Type** values are combined into the Type attribute in Schema Out.
- The Data Quality **Content_Type** migrates to the Content Type attribute in Schema Out.



Ignored flat file Reader options

The `dqmigration` utility ignores the following flat file Reader option groups.

Option group	Data Quality flat file Reader option	Result of dqmigration utility
Transform_Performance	Bulk_Read_Size_Recs	Ignored
	Buffer_Size_Bytes	Ignored
SQL_Options	None apply to flat files.	Ignored

7.4.3.3.2 How flat file Writer transforms migrate

The `dqmigration` utility migrates Writer transforms with flat file outputs to the following Data Services objects:

- File Format -- A file format is a set of properties describing the structure of a flat file (ASCII). File formats describe the metadata structure which can be fixed or delimited.
- Query transform -- A Query transform maps the fields from the previous transform to the output fields in the following transform or target.
- Target -- A target is an object into which Data Services loads data. In this case, it is the specific flat file into which the migrated Data Quality Writer transform loads data.

The following topics describe the mapping of the Data Quality flat file Writer options to the Data Services Query and Target options.

- Mapping of flat file Writer options
- Example of migrated flat file Writer
- Output difference when load to delimited file
- Migration of delimited files
- Performance improvement suggestions for flat file Reader and Writer transforms

Mapping of flat file Writer options

The migration of flat file Writer transforms is very similar to that of flat file Reader transforms.

- The Data Quality Connection_Options migrate to connection options in the Data Services file format for the target.

- The Native_Flat_File_Options migrate to additional options in the file format.
- The Output_Field_Options migrate to the Data Services Query transform.

The `dqmigration` utility ignores options in the following Writer transform option groups:

- Transform_Performance
- Once_Per_Writer_SQL_Options
- Per_Write_SQL_Options
- MIsc_Options

Related Topics

- Migration of flat file Reader Connection_Options
- Mapping Native_Flat_File_Options
- Example of migrated flat file Writer

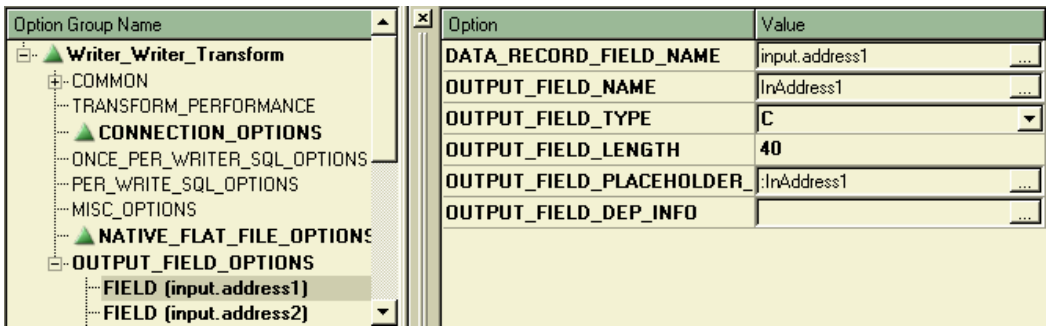
Example of migrated flat file Writer

This example shows how the `dqmigration` utility migrates a Data Quality project with a flat file Writer to Data Services.

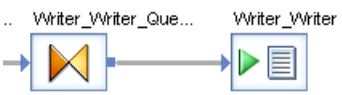
Suppose the Data Quality project `my_address_cleanse_usa.xml` has a Writer with the following Connection_Options values:

- **Data_Source_Name**: `$$SAMPLE_OUTPUT_DATA_PATHaddress_cleanse_usa.csv`
- **Driver_Name**: `FLDB_DELIMITED`

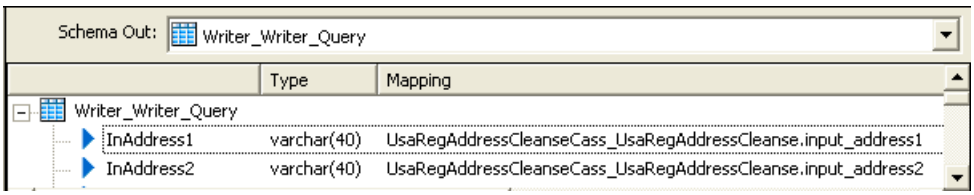
Suppose this Writer also has the following output fields.



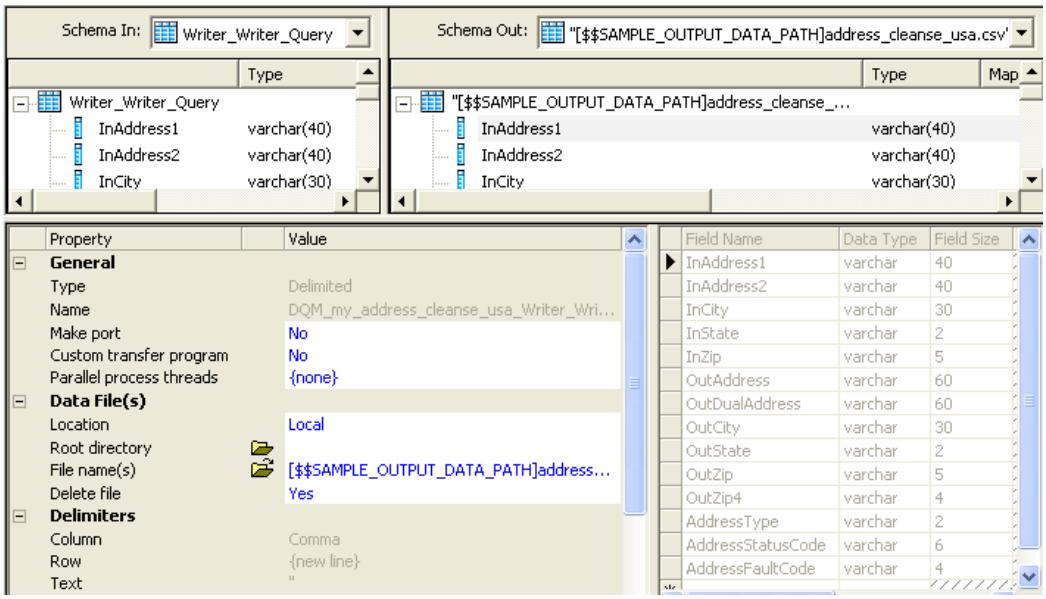
The `dqmigration` utility migrates this Writer to the following two Data Services objects in a data flow:



- The Writer_Writer_Query transform contains the mapping of the fields in the upstream transform to the target file format. In this example, the Schema Out in the Writer_Writer_Query transform shows that:
  - The target column `InAddress1` maps from the upstream field `input_address1`.
  - The column `InAddress1` has a **Type** `varchar(40)` which the `dqmigration` utility derives from the Data Quality **Output_Field_Length** and **Output_Field_Type** options.

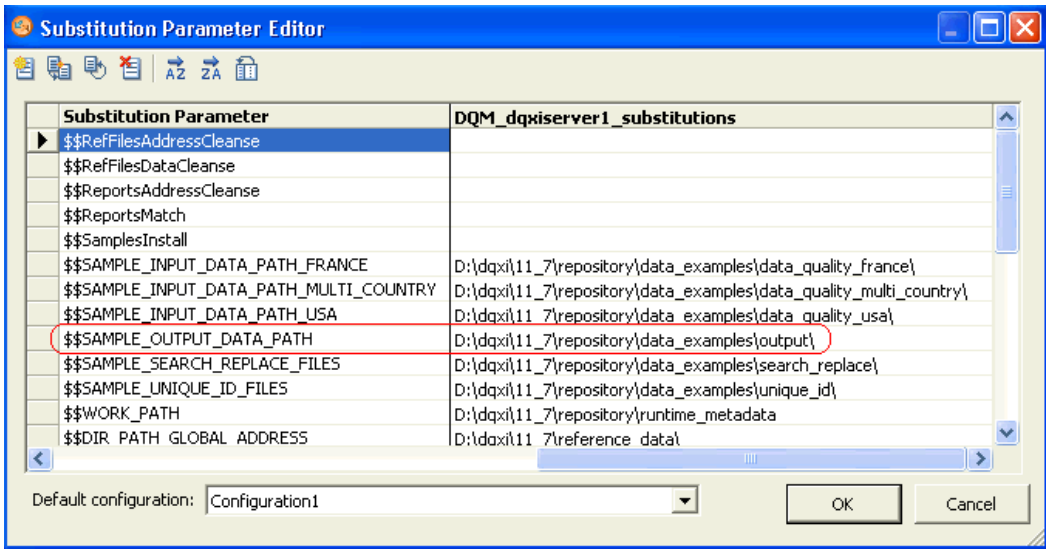


- The following file format for the target Writer_Writer shows these migrated values:
  - **Name**: `DQM_my_address_cleanse_usea_Writer_Writer`
  - **Type**: `Delimited`
  - **File name(s)**: `$$SAMPLE_OUTPUT_DATA_PATHaddress_cleanse_usa.csv`



File name(s) contains a substitution parameter. To view the value of substitution parameters, click **Tools > Substitution Parameter**.

The following Substitution Parameter Editor shows the value of `$$SAMPLE_OUTPUT_DATA_PATH`.



Output difference when load to delimited file

Data Services behaves differently from Data Quality when no data exists in a field.

- Data Quality loads double quotes "" when no data exists in a field. Data Quality does not differentiate between NULL and an empty string.
- Data Services loads nothing if the value is NULL. If the data is empty, Data Services loads double quotes "".

For example, suppose the data before columns "Nyc" and "HD" are NULL.

- Data Quality loads the following values:

```
 "14 60TH STREET","","NYC","NY","10022",
"14 E 60th St","","Nyc","NY","10022","","HD",
"S20000","E601"
```

- Data Services loads the following values:

```
 "14 60TH STREET",,"NYC","NY","10022",
"14 E 60th St",,"Nyc","NY","10022",,"HD",
"S20000","E601"
```

Related Topics

- Migration of delimited files

7.4.3.3.3 Migration of delimited files

In Data Quality, delimited file formats (with the `.dmt` suffix) contain parameters for delimited files. The `dqmigration` utility uses these file formats to define the Data Services File Format.

The following table shows how the `dqmigration` utility maps the delimited file parameters to the Data Services File Format options.

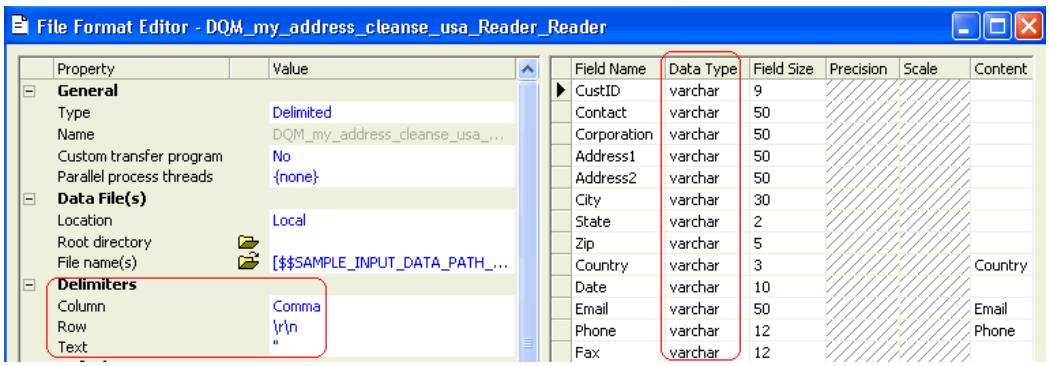
Note:
If the .dmt file contains empty strings for any of the parameters, Data Services substitutes the default value for the option.

Data Quality delimited file parameter	Data Services File Format section and option	Comments
FIELD DELIMITER	**Delimiter** section, **Column**	In Data Quality, this parameter contains the ASCII value. The Data Services option contains the actual character. For example, if the ASCII value is `9`, the Data Services value is `\t` . If no value is specified in the Data Quality parameter, Data Services uses the default value `comma`.
RECORD DELIMITER	**Delimiter** section, **Row** If the .dmt file contains `'13 10'` , **Row** will contain `'\n'`.	In Data Quality, this parameter contains the ASCII value. In Data Services, this option contains the actual character. For example, if the ASCII value is `10`, the Data Services value is `\n`. If no value is specified in the Data Quality parameter, Data Services uses the default value `{new line}`.
FIELD FRAMING CHARACTER	**Delimiter** section,**Text**	Data Services allows only single characters as the text delimiter. The most common character is a double quote or a single quote. If the Data Quality parameter contains a number, the `dqmigration` utility converts it to a character. If no value is specified in the Data Quality parameter, Data Services uses the default value `(")`.
Actual Fields, such as CustID, Contact, Corporation	These fields become the Field Names in the schema for the file format.	Data Services specifies the length of each column. If the length is not available in the .dmt file, the Field Size is set to `1024` and the Data Type to `varchar`. Data Services does not allocate 1024 bytes, but allocates the optimum value based on the data.

For example, suppose your Data Quality .dmt file has the following contents with ASCII values for the delimiters and each field has data type `c`.

```
FIELD DELIMITER = 44
RECORD DELIMITER = 13 10
FIELD FRAMING CHARACTER = 34
CustID,9,c
Contact,50,c
Corporation,50,c
Address1,50,c
Address2,50,c
City,30,c
State,2,c
Zip,5,c
Country,3,c
Date,10,c
Email,50,c
Phone,12,c
Fax,12,c
```

After migration, the resulting File Format in Data Services contains character values for the delimiters and the data type is `varchar` as in the following screen shot shows.



Related Topics

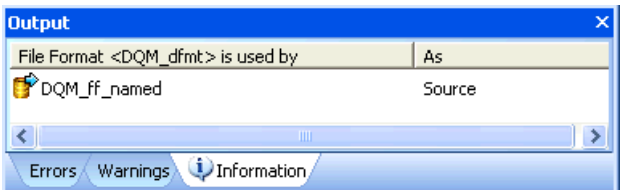
- Migration of flat file Reader Connection_Options
- Mapping of flat file Writer options

7.4.3.3.4 Performance improvement suggestions for flat file Reader and Writer transforms

The `dqmigration` utility creates a file format for each flat file Reader and Writer in each Data Quality project, which might result in multiple flat files with the same connection information. To improve performance, consolidate the file formats and reuse a file format in multiple data flows as follows:

1. Determine which data flows use file formats with the same connection information.
   a. In the Data ServicesDesigner, go to the File Formats tab in the Local Object Library.
   b. Open each migrated file format to see which ones have the same **File name(s)** value.
2. Choose one file format to reuse and which duplicate file formats to replace.
3. Open each data flow that uses a duplicate file format and replace it with the file format you chose to reuse.
   a. In the Local object Library, select the file format name, **right-click > View Where Used**.

The Output window shows the name of each data flow that uses the file format and indicates that the file format is used as a Source or as a Target. For example, the following Output window shows that the `DQM_fmt` file format is used in data flow `DQM_ff_named` as Source.



b. Double-click the name of the data flow.

   The data flow opens in the workspace with the Source or Target highlighted.

c. Delete the high-lighted Source or Target.

d. From the Local Object Library, select your chosen file format to reuse and drag it into the workspace, connect it to the data flow, save the data flow and close it.

4. In the Local Object Library, verify that the Usage column for each duplicate file format has the value `0`. If it is non-zero, repeat the previous step to replace the duplicate file format within the data flow where it is used.
5. Delete the duplicate file format(s).

Related Topics

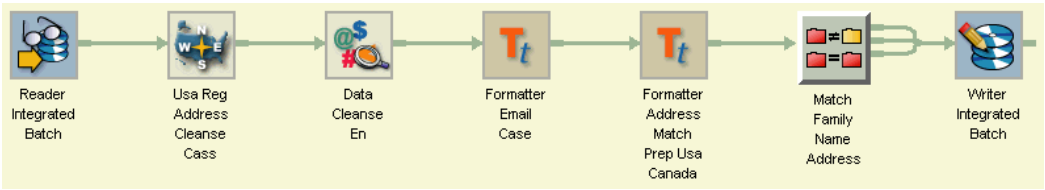
- Migration of flat file Reader Connection_Options
- Mapping of flat file Writer options

7.4.4 How Data Quality integrated batch Readers and Writers migrate

7.4.4.1 Migrating Data Quality integrated batch projects

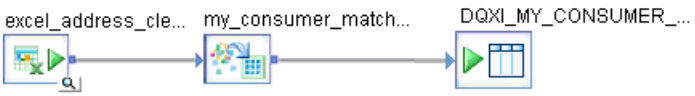
If the imported Data Quality integrated batch project in your Data Integrator 11.7 data flow contains a large number of transforms, it would be easier to replace the Reader and Writer in the migrated Data Quality integrated batch project than to rebuild the Data Quality project by adding and configuring the necessary Data Services transforms.

For example, suppose your Data Quality integrated batch project has many transforms such as the following.



Further suppose that this Data Quality project, `my_integ_batch_address_cleanse_usa`, was used in a Data Integrator 11.7 data flow that uses the following source and target objects:

- excel_address_cleanse_source
- DQXI_MY_CONSUMER_target

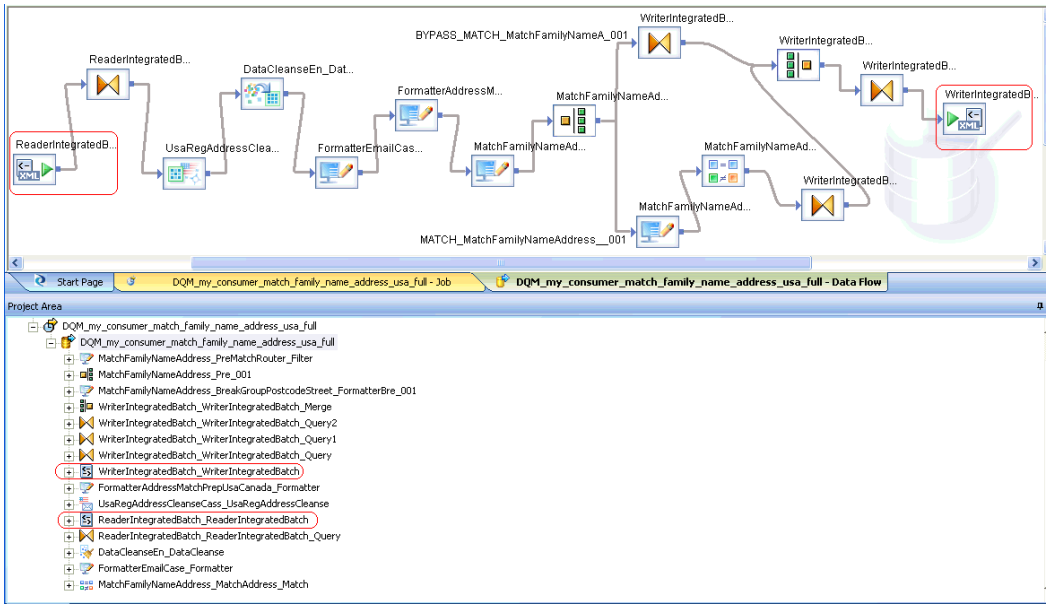


Similar to a Data Quality batch project, the `dqmigration` utility migrates a Data Quality integrated batch project to two Data Services objects: a Data Flow and a Job. However, you must take additional actions to run the migrated job in Data Services.

To replace the migrated Readers and Writers with the appropriate sources and targets, perform the following steps.

1. After you run the `dqmigration` utility on your Data Quality repository, open the migrated integrated batch job and data flow in the Data ServicesDesigner.

   Using the above example, the resulting migrated data flow looks like the following in Data Services.



The `dqmigration` utility migrates this integrated batch project to a Data Services job named DQM_my_integ_batch_address_cleanse_usa which contains a data flow with the same name. The `dqmigration` utility adds a `DQM_` prefix to your project name from Data Quality to form the Data Services job and data flow name.

2. Replace the placeholder XML Reader with the source that the Data Integrator 11.7 data flow used.
   a. Disconnect the placeholder XML Reader (circled on the left in the above data flow).
   b. Go to the Local Object Library and drag the source that the Data Integrator 11.7 data flow used into the workspace. The source can be either a file or a table in a datastore.

      Using the above example, drag the source excel_address_cleanse_source into the workspace.

   c. Connect the source to the migrated Reader_Query transform and ensure that the mapping of columns in Schema In to columns in Schema Out in the Reader_Query are correct.
3. Replace the placeholder XML Writer with the Target that the Data Integrator 11.7 data flow used.
   a. Disconnect the placeholder XML Writer (circled on the right in the above data flow).
   b. Go to the Local Object Library and drag the target used by the Data Integrator 11.7 data flow into the workspace. The target can be either a file or a table in a datastore.

      Drag the target DQXI_MY_CONSUMER_target into the workspace.

   c. Connect the target to the migrated Writer_Query transform. Open the transform and ensure that the mapping of columns is correct for the Schema In and Schema Out columns.
4. Delete the placeholder XML Reader and placeholder XML Writer from the data flow.

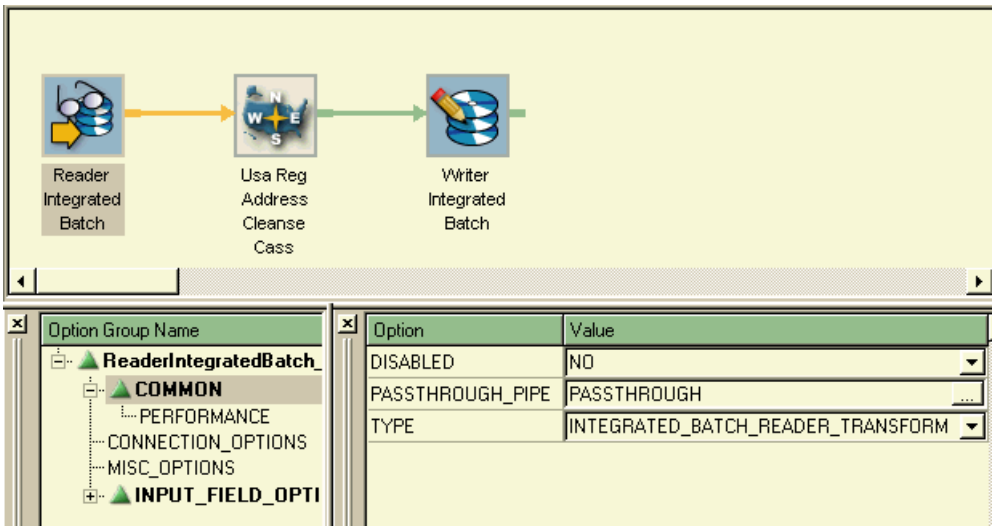
Related Topics

- Running the dqmigration utility
- How integrated batch projects migrate

7.4.4.2 Modifying Data Integrator 11.7 Data Quality projects

If your integrated batch project in a Data Integrator 11.7 data flow contained one or two Data Quality transforms, it would be easier to modify your data flow to replace the imported Data Quality transforms with the corresponding Data Quality transforms in Data Services.

For example, suppose your Data Quality integrated batch project has only one transform such as the following.

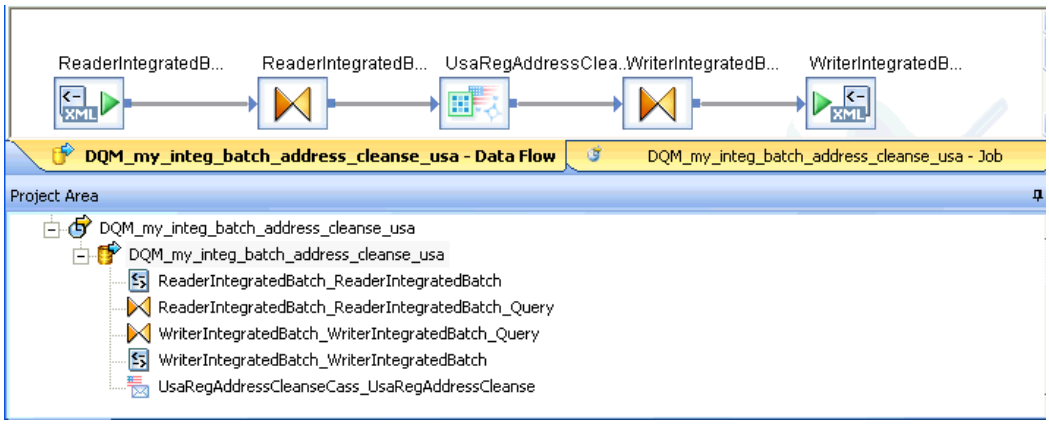


Perform the following steps to modify your existing Data Integrator 11.7 data flow that contains a Data Quality project to make it operable in Data Services 12.0.

Note:
Steps 1 and 2 are optional. Perform them only if you need to refer to the mapping in the migrated integrated batch project to setup the same mapping within the new Data Quality transform in step 3.

1. Run the `dqmigration` utility on your Data Quality repository.
2. Log in to the Data ServicesDesigner and open the migrated integrated batch job to view the mappings in the transform.

   Using the above example, the resulting migrated data flow looks like the following in Data Services.

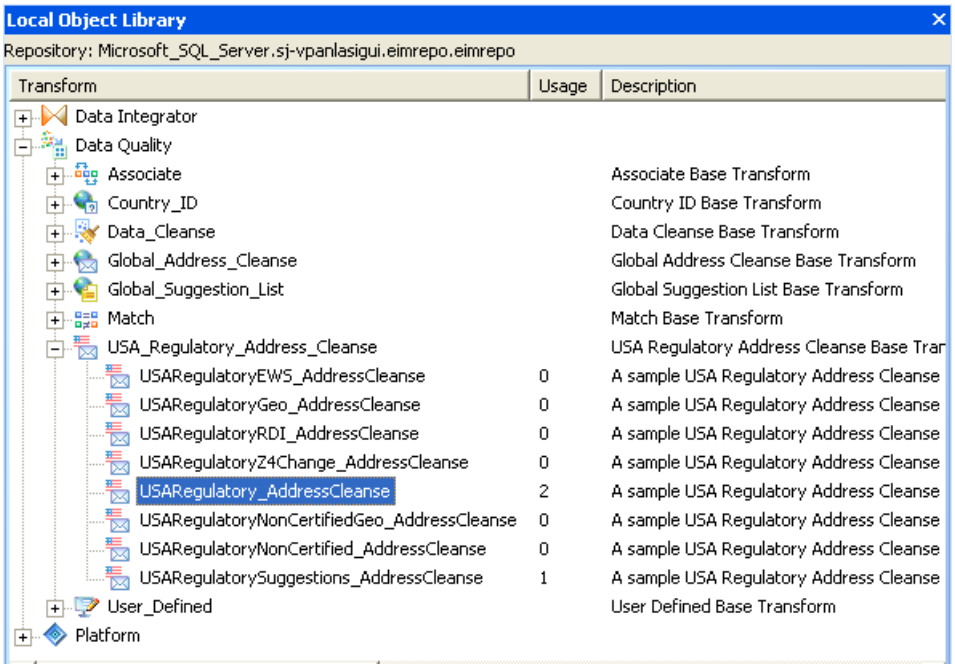


   In this example, the migrated project is not used in Data Services. Only the mapping in the migrated UsaRegAddressCleanse transform is used as a reference to map the input and output fields of the Data Quality transform in Data Services.

3. In the original 11.7 data flow, replace the imported Data Quality project with the new Data Services Data Quality transform.

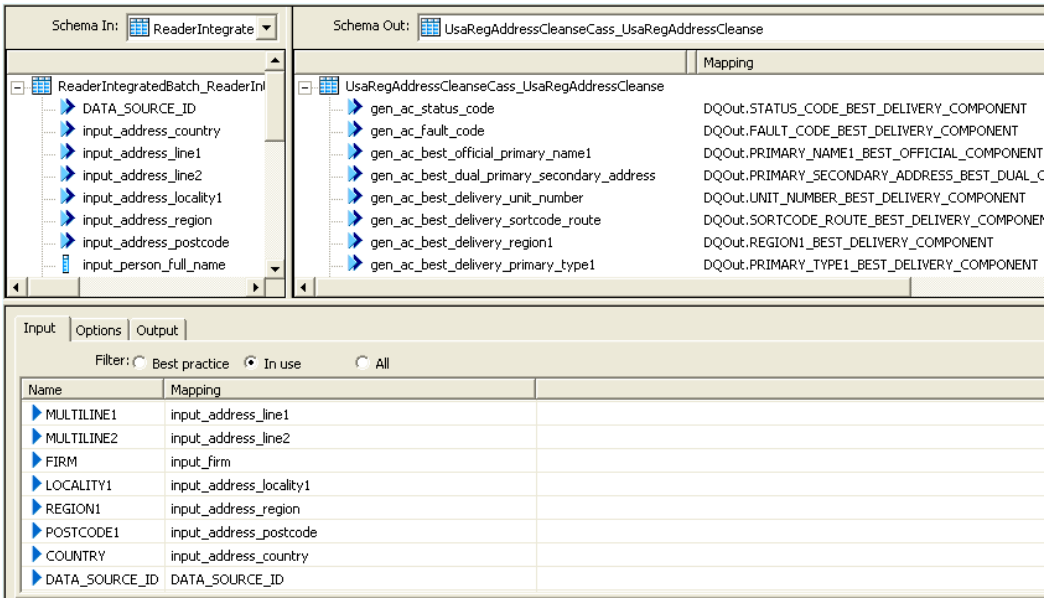
a. Open your existing data flow that contains a Data Quality project.
b. In the Transform tab of the Local Object Library, find the equivalent Data Quality transform.

   In the example, the imported Data Quality Usa Reg Address Cleanse Cass now has the equivalent USARegulatory_AddressCleanse in Data Services that you can drag from the Local Object Library to your data flow.



c. Refer to the mapping in the migrated integrated batch project to setup the same mapping within this Data Quality transform.

   In the example, the mapping within the UsaRegAddressCleanseCass_UsaRegAddressCleanse transform looks like the following.



Related Topics

- How integrated batch projects migrate

7.4.5 How Data Quality transactional Readers and Writers migrate

7.4.5.1 How transactional Readers and Writers migrate

The `dqmigration` utility migrates Data Quality transactional Reader transforms to the following Data Services objects:

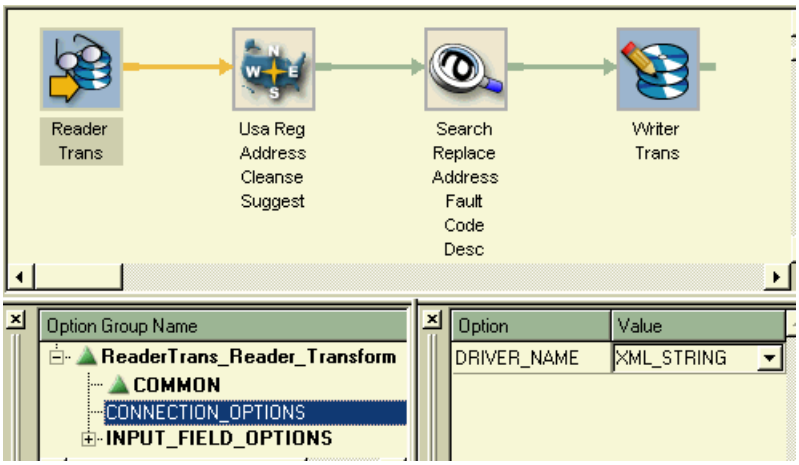
- XML message -- Indicates a real-time source or target in a job.
- Query UnNest transform -- Unnests the fields from the XML message into a flat list of fields.
- Query Reader transform -- Maps the unnested fields as input fields to the next transform.
- Row_Generation transform -- Drives processing for multiple input records in the same message. It contains one column, `DI_ROW_ID`.
- Query Driver transform -- Drives processing for multiple input records in the same message. It contains one column, `dataflow_id`, mapped to the job name.

The `dqmigration` utility migrates Data Quality transactional Writer transforms to the following Data Services objects:

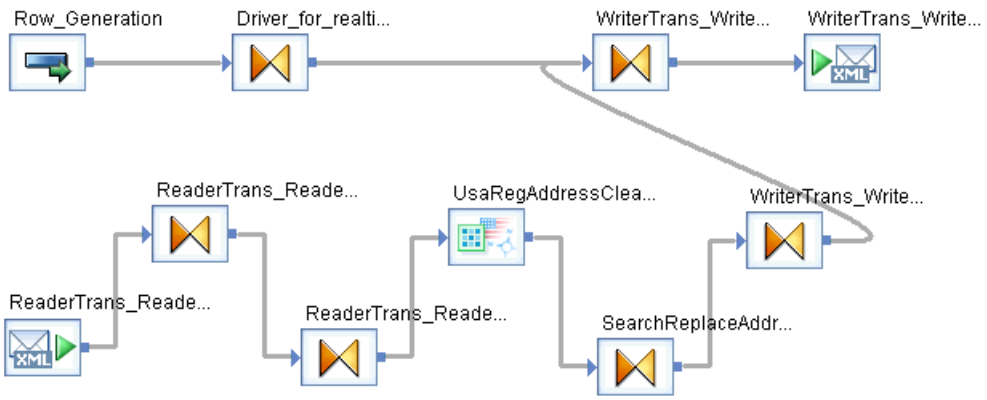
- Query Nest transform -- Reinstates the nesting of the records before they are loaded into the XML message target.
- XML message target -- Indicates a real-time target in a job.

7.4.5.2 Example of transactional Reader and Writer migration

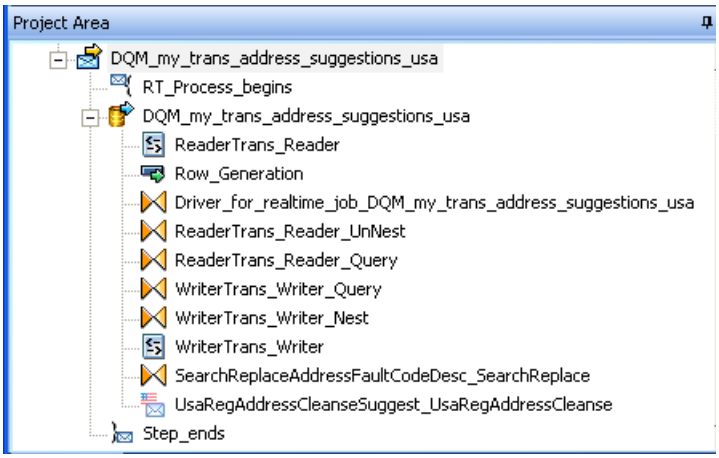
Suppose you have the following Data Quality transactional project, `my_trans_address_suggestions_usa.xml`, that has the following Connection_Options values. The **Driver_Name** value `XML_STRING` indicates that this Reader transform is a transactional Reader.



The `dqmigration` utility migrates this Data Quality transactional project to the following Data Services data flow.



The Project Area shows the full names of these transforms.



The data flow, DQM_my_trans_address_suggestions_usa, contains the following migrated transforms:

Data Quality transform	Data Services objects after migration
Transactional Reader transform `ReaderTrans`	Source object, `ReaderTrans_Reader`, which reads data from file format named `DQM_my_trans_address_suggestions_usa_Reader Trans_Reader`
	Query transform, `ReaderTrans_Reader_UnNest`, which unnests the record from the file format to the next transform
	Query transform, `ReaderTrans_Reader_Query`, which maps the fields from the unnesting query to the next transform
	Row_Generation transform which drives processing for multiple input records in the same message
	Query transform, `Driver_for_realtime_job_DQM_my_trans_ad dress_suggestions_usa` , which drives processing for multiple input records in the same message
`UsaRegAddressCleanseCass`	`UsaRegAddressCleanseCass_UsaRegAddressCleanse`
`SearchReplaceAddress FaultCodeDesc`	Query transform, `SearchReplaceAddressFaultCodeDesc_SearchRe place` , which uses the SearchReplace function in the column mapping

Data Quality transform	Data Services objects after migration
Transactional Writer transform `WrierTrans`	Query transform, `WriterTrans_Writer_Query`, which maps the fields from the previous Query transform
	Query transform, `WriterTrans_Writer_Query_Nest`, which reinstates the nesting of the records before they are loaded into the target message
	Target object, `WriterTrans_Writer`, which loads data into file format named `DQM_my_trans_address_suggestions_usa_Writer Trans_Writer`

Related Topics

- How transactional projects migrate
- How transactional Readers and Writers migrate

7.4.5.3 Post-migration tasks for transactional Readers and Writers

After migrating a Data Quality transactional project, you must publish the resulting Data Services real-time job as a Web service. Perform the following steps to generate a real-time job in Data Services.

1. In the Data ServicesServer Manager, create an Access Server to manage the real-time communication between Data Services and the external web application that is sending each transaction.
2. In the Administrator in the Data ServicesManagement Console, perform the following tasks:
   a. Add a connection to the Access Server using the Management node.
   b. Create a service to process requests for the real-time job using the Real-Time node. Also create the service provider to perform that service.
   c. Start the services.

Related Topics

- How transactional projects migrate
- Management Console Guide: Administrator, Real-Time Jobs

7.4.6 Matching transforms

7.4.6.1 How Match transforms migrate

The match process in Data Quality XI used a number of different transforms to perform the necessary processing. In Data Services, you will use only two transforms, to do the bulk of the match processing: the Match and Associate transforms.

The `dqmigration` utility migrates Data Quality match-related transforms to the following Data Services objects:

Transform	Migrates to...
Match	Match transform. If you had more than one Match transform in your project, the migration utility determines if the transforms were being used as match levels, match sets, or both. Each match set creates a Data Services Match transform. A Data Services Match transform can now contain multiple match levels.
Group Statistics	Match transform: Group statistics operation for each migrated match level.
Best Record	Match transform: Best record operation for each migrated match level.
Unique ID	Match transform: Unique ID operation for each migrated match level.
Associate	Associate transform
Candidate Selector	Match transform: Candidate selection Group Forming operation.
Formatter/Sorter/Aggregator (Break group compound transform)	Formatter transform: Migrates to a User-Defined transform. Sorter/Aggregator: Migrates to a Break group operation in the Match transform (Group Forming operation).
Source	User-Defined transform

7.4.6.2 Match project patterns

One of the ways the migration utility determines how to create a new Data Services data flow is by recognizing patterns (or sequences of transforms) in your projects, particularly in your matching-related transforms.

While you may have many different transforms in your Data Quality project, in all different positions in your dataflow, there are some guidelines for setting up a match project to perform the needed functions at the appropriate times. For example, if you want to collect statistics about your match groups, it is necessary to have a Group Statistics transform after your Match transform. Or, if you want to control which records are the master record in a match group, you might place a per-collection Sorter transform before your Group Statistics transform.

In general, if you do not follow the usual patterns, the migration utility will probably create multiple Match transforms where a single transform could have been created.

When these patterns are not followed, it is more likely that you will be required to perform post-migration changes to your data flows.

7.4.6.2.1 One or many migrated Data Services Match transforms?

The biggest difference between Data Quality XI and Data services match functionality is that Data Services can accomplish with one Match transform what took many transforms to accomplish in Data Quality XI. However, this does not necessarily mean that you will always see fewer Match transforms in your migrated project. The pattern of the transforms in your Data Quality match project plays a role in how it migrates.

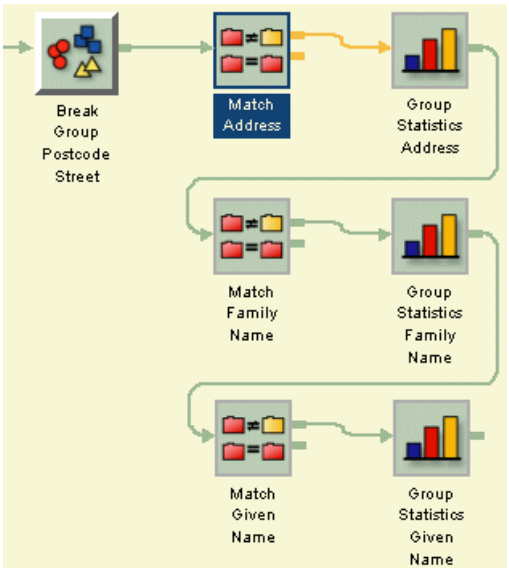
In general, you see a migrated Match transforms for one or several of the following reasons:

- It contains group forming (break groups, candidate selection, and so on) operations.
- It contains matching operations (match levels, criteria, and so on).
- It contains post-match operations (group statistics, best record, unique ID, and so on).

Example: **Match levels: Many to one**

One of the more common scenarios you will find involves match levels. In Data Quality, each Match transform could represent one match level. In Data Services, each Match transform can hold multiple match levels.

In Data Quality, the Consumer Householding blueprint contained a Match compound transform that contained three Match transforms and three corresponding Group Statistics transforms.



In Data Services, these transforms are migrated into one Match transform, with the levels included and their corresponding Group Statistics operations.

Example: One to many

There may be instances where you have only one Match transform in your Data Quality, but you also have some match-family transforms in that dataflow. In these cases, you may have only one or you may have several Match transforms. What you end up with depends on whether your project fits the pattern of "normal" match dataflows.

Related Topics

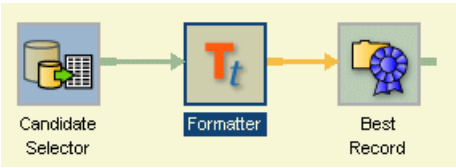
- Broken match patterns

7.4.6.2.2 Broken match patterns

There are a few scenarios where the migration utility might have trouble recognizing what you attempting to accomplish in a particular area of your project. In these cases, you may need to do some manual corrections in your data flow.

Therefore, if your project does not contain any other extraneous transforms (other than a Sorter or Group Statistics transform, for example) in between your Match transforms, your Data Services data flow should contain only one Match transform to account for all match levels.

For example, suppose you have the following transforms as part of your match project.



Because the Data Services Match transform contains both a candidate selection and best record operation, you might think that this would migrate into just one Match transform.

However, this section of the project is migrated to two Match transforms with a User-Defined transform separating them, because the sequence of transforms that would allow a project to be migrated into one Match transform has been broken.

The new Data Services Match transform does not allow for special formatting of data in between matching operations. You must perform this formatting of data before it enters the Match transform.

If the Formatter transform, in the example, did not exist, the Candidate Selector and Best Record transforms could be migrated and merged into one transform, containing only candidate selection and best record operations.

Tip:
After migration, you may want to consolidate your separate Match transforms to make editing them easier. However, you will need to be careful about re-mapping fields mapping and moving other transforms around that may have performed an operation that you want to use in one of the Match transforms.

Also, Match transforms that are created due to broken patterns may need their Group Forming operations filled in, based on what existed in the corresponding Data Quality project.

Related Topics

- How to set up Group Forming section for Match

7.4.6.2.3 Match transform input and output pipes

What happens when there are multiple input pipes to a Match transform that are connected to upstream transforms that are within the pattern?

Generally, such a condition will result in breaking of the "usual" pattern and we will not be able to migrate that pattern into a single Match transform, even though the sequence of the transforms is "usual". You will probably see multiple Match transforms.

Multiple output pipes

In Data Quality, a Match transform could have multiple output pipes like MATCH_GROUP_PIPE, UNIQUES_PIPE, and so on. In Data Services, Match can have only one output pipe, and all incoming records are available on output. To simulate the multiple output pipes during migration, we add a Case transform after the Match transform. The conditions in that Case transform work on the Group Number output field of the Match transform to segregate the unique and duplicate records when needed.

Note:
If in the original Data Quality project, all of the output pipes of the Match transform are connected to the same downstream transform, then after migration, there will be no Case transform since there is no need to segregate the records.

Multiple input pipes

What happens when there are multiple input pipes to a Match transform that are connected to upstream transforms that are within the pattern?

Generally, such a condition will result in breaking of the "usual" pattern and we will not be able to migrate that pattern into a single Match transform, even though the sequence of the transforms is "usual". You will probably see multiple Match transforms.

7.4.6.2.4 Group numbers

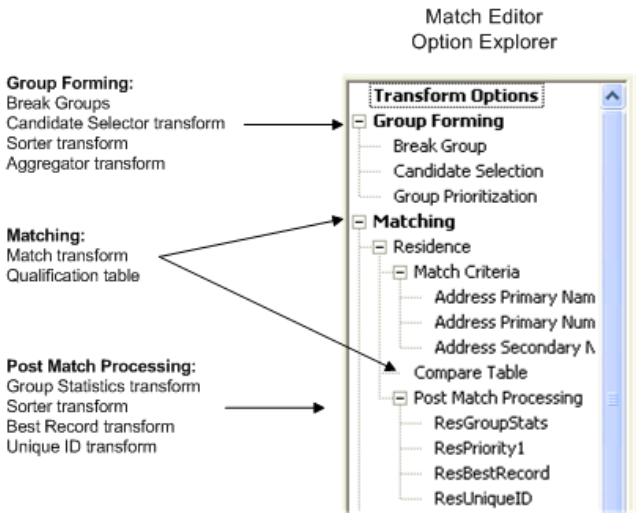
Any User-Defined, Group Statistics, Best Record, or Unique ID transform that migrates to a separate Data Services Match transform will use the group number field, generated in the upstream Match transform for break group creation.

7.4.6.3 Migration of match options

Most of the Data Quality match-related transforms, such as the Match, Group Statistics, Best Record, and Unique ID transforms migrate to a single Match transform.

The Match Editor, a new user-interface for setting Match transform options, provides you with an easy way to set up even the most complex matching in your data flow.

Options in the Data Services Match transform are divided into three main categories or operations: pre-match operations (Group Forming), such as break group creation and candidate selection; matching; and post-match operations, such as best record, group statistics, and group prioritization, which are all available for each match level defined in the Match transform.



7.4.6.4 Matching terminology and definition changes

With the migration to Data Services, you will also notice some changes to terminology. In some cases, terms changed to make them more intuitive. In other cases the definition of a term has a slightly different meaning in Data Services.

Data Quality term	Data Quality description	Data Services term	Data Services description
Match set	A match transform used to process a particular set of data.	Match set	No change.
Match level	A match transform used to process a particular type of data, such as individual, family, or residential data. Each level was processed by a single Match transform.	Match level	A Match transform can now process multiple levels.
Standard or custom keys	Used to help the Match transform identify the contents of a particular field so that it could perform some data optimization for matching purposes.	Criteria	You now choose from among a list of criteria names, with "Custom" being one of the options.
Data Source ID	Value that identifies the origin of the records being processed. (A Reader transform)	Physical source value	

Data Quality term	Data Quality description	Data Services term	Data Services description
Source ID	Value that identifies a subset or a superset of records.	Logical source value	
Qualification table	Table of rules for comparing, or not comparing, records based on the source	Compare table	
Best record priority	A score given to a record to allow you elevate it's importance in a break group or match group.	Record priority	
Best record filter	Python code set up to determine whether to perform an action based on a True or False value	Strategy	Same as a filter, although the Best Record operation now has some pre-made strategies for you to choose from, derived from common use cases.

7.4.6.5 Deprecated features

Standard keys

The following standard match keys (now referred to as criteria) have been removed from Data Services. These standard keys are migrated as different criteria.

Data Quality standard key	Data Services criteria
ADDRESS_LINE	ADDRESS_DATA1
FIRM_LINE	FIRM_DATA1
LAST_LINE	ADDRESS_DATA1
NAME_LINE	NAME_DATA1
UNPARSED_ADDRESS_LINE	ADDRESS_DATA1
UNPARSED_LAST_LINE	ADDRESS_DATA1

If you want, you can change or remove these in the migrated Match transform.

Output fields in Best Record operation

You can no longer create a new output field in Best Record. You must post to existing input fields.

Candidate selection

- The option, EXECUTE_SQL_PER > ALL_RECORDS_IN_COLLECTION, no longer exists.
- You can longer create a unique record ID from the candidate selection functionality.

Associate transform

The INPUT_MODE > Transaction option value is not supported. It is not applicable anymore.

7.4.7 UDT-based transforms

7.4.7.1 How User-Defined transforms migrate

The `dqmigration` utility migrates Data Quality User-Defined transforms to User-Defined transforms in Data Services. It also migrates other Data Quality transforms that are based on the Data Quality User-Defined transform to Data Services. The following Data Quality transforms are migrated to the User-Defined transform in Data Services:

- User-Defined
- Filter
- Formatter
- Scan and Split

Other transforms derived from the Data Quality User-Defined transforms are migrated to other transform types:

- Copier is migrated to a Query transform
- List/Source is migrated to a Query transform

If you have a Data Quality User-Defined or Filter transform with multiple output pipes, the migration utility makes several modifications to the transform and data flow. You must also manually modify the Data Services data flow. For more information, see Multiple pipe support.

Although the User-Defined transform is flexible and powerful, many of the tasks that you may want to perform can be also accomplished with the Query transform. The Query transform is generally more scalable and faster, and uses less memory, than User-Defined transforms.

Related Topics

- How User-Defined transforms migrate
- Performance suggestions for User-Defined transforms
- Multiple pipe support

7.4.7.1.1 Migration of User-Defined transform options

The following tables show how the `dqmigration` utility maps important Data Quality User-Defined transform options to the Data Services User-Defined transform options. The rest of the options are ignored.

Options

Data Quality User-Defined transform option	Description	Data Services User-Defined option
Expression	Specifies the Python expression for the transform.	Any pre- and post-expression that was previously added during runtime in Data Quality will be now added explicitly to the Python expression. See "How Python expressions migrate."
Processing_Mode	Specifies the logic of the Expression setting.	If Processing_Mode is Per_Record, the value remains the same in Data Services. If Processing_Mode is Per_Collection, a Data_Collection_Config option group is created, which must be manually entered. A warning is issued. If Processing_Mode is Per_Dataflow, a warning is issued. Per Dataflow mode is not supported in Data Services. You can rewrite the expression using Data Services scripting language.

For information about how output pipes migrate in User-Defined transforms, see Multiple pipe support.

Related Topics

- How Python expressions migrate
- Executing per data flow scripts using Data Services scripting language

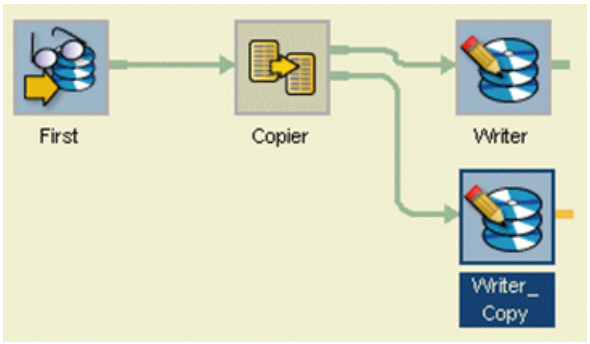
Copier transform

In Data Quality, a Copier transform is used to copy all of the input rows and send them into a number of pipes.

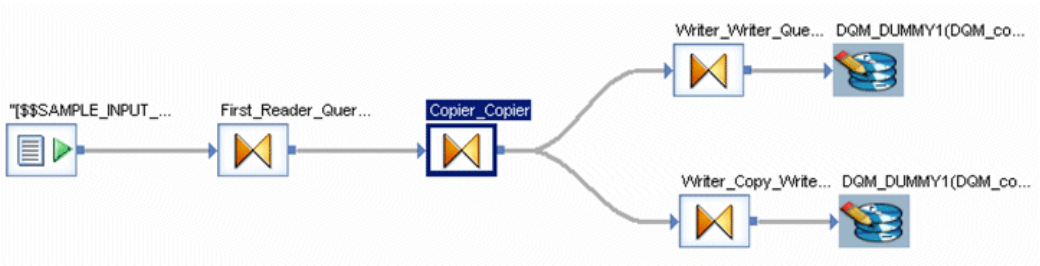
Even though the Copier transform is based on the User-Defined transform, it does not need to be migrated as such because the functionality can be achieved differently in Data Services. In Data Services, a Copier transform is migrated to a Query transform and the output schema of the Query transform is joined to the corresponding transforms.

Generally, there are no input or output fields in Copier transforms, so the schema of the Query transform consists of pass-through fields with straight-forward mapping. If any input or output fields exist in the Copier transform, the migration utility logs a warning and ignores those fields.

For example, suppose you have a Data Quality project that includes a Copier transform that has two output pipes to two Writer transforms.



The `dqmigration` utility creates a Data Services data flow with a Query transform.



Filter transform

The Data Quality Filter transform is based on the User-Defined transform, and in the migration to Data Services, it undergoes the same modifications as the User-Defined transform.

Generally, there are no input or output fields in the Filter transform, so the schema of the Data Services User-Defined transform consists only of pass-through fields with straight-forward mapping. If any input or output fields exist in the Filter transform, the migration utility logs a warning and ignores those fields.

If you have a Data Quality Filter transform with multiple output pipes, the migration utility makes several modifications to the transform and data flow. You must also manually modify the Data Services data flow. For more information, see Multiple pipe support.

Related Topics

- Multiple pipe support

Formatter transform

The Data Quality Formatter transform is based on the User-Defined transform, and in the migration to Data Services, it undergoes the same modifications as the User-Defined transform.

Scan and Split transform

The Data Quality Scan and Split transform is based on the User-Defined transform, and in the migration to Data Services, it undergoes the same modifications as the User-Defined transform.

7.4.7.1.2 Performance suggestions for User-Defined transforms

After migration, all of your Data Quality User-Defined transforms (and Filter, Formatter, and Scan and Split transforms derived from the User-Defined transform) are migrated to Data Services User-Defined transforms.

Although the User-Defined transform is flexible and powerful, many of the tasks that you may want to perform can be also accomplished with the Query transform. The Query transform is generally more scalable and faster, and uses less memory, than User-Defined transforms.

To improve performance, especially if the Python expression in your Data Quality transform is simple, you may consider replacing your User-Defined transform with a Query. For a Filter transform, a Case transform may be more appropriate.

You can also rewrite some of your more complex transforms as a custom function and call it from within a Query transform. Custom functions are written in the Data Services scripting language. Because you can run custom functions in parallel, you can expect better performance than with Python.

For more information about the Query transform and adding it to your data flow, see the *Reference Guide* and *Designer Guide*.

7.4.7.2 How Python expressions migrate

The `dqmigration` utility migrates Data Quality Python expressions to Data Services.

Related Topics

- Migration of Python classes and methods
- Reference Guide: Python in Data Services

7.4.7.2.1 Migration of Python classes and methods

The `dqmigration` utility migrates Data Quality Python expressions to Data Services. However, some of the classes and methods that were available in Data Quality are not available or are deprecated in Data Services.

The classes and methods that are no longer supported are those that can be accomplished in other ways in Data Services, are unnecessary in the Data Services environment, or were rarely used. After migration, Python expressions containing these classes and methods will fail at runtime and must be fixed manually.

Although deprecated methods are technically supported in Data Services, they are not displayed in the Python Expression editor, are not documented, and they may be removed in a future Data Services release.

Class	Method	Status in Data Services
FIDataCollection	AddRecord	Supported
	DeleteRecord	Supported
	GetCurrentStatus	Not supported
	GetRecord	Supported
	InvalidateIterator	Not supported
	SaveRecord	Deprecated
	SendToPipe	Deprecated
	Size	Supported
	Truncate	Supported
FIDataManager	AddCollection	Not supported
	DeleteDataCollection	Not supported
	DeleteDataRecord	Supported
	NewDataCollection	Not supported
	NewDataRecord	Supported
FIDataRecord	GetField	Supported
	GetFieldsXML	Not supported
	GetNumFields	Not supported
	HasField	Not supported

Class	Method	Status in Data Services
	SendToPipe	Deprecated
	SetField	Supported
FIProperties	GetProperty	Supported
FIPythonString	GetBuffer	Supported
	Set Buffer	Supported
FIRemoveService	RunBatchDataflow	Not supported

Related Topics

- Reference Guide: Python

7.4.7.2.2 Migration of Python expressions

GetField and SetField methods use Mapped_Name

In Data Quality, the GetField and SetField methods use the Data_Record_Field_Name. In Data Services, both methods use Mapped_Name. However, they are backward compatible, so that migrated User-Defined transforms containing these methods can still use Data_Record_Field_Name.

During migration, the value of the Data_Record_Field_Name in the Python expression is replaced with the value of the Mapped_Name.

For example, if there is a Data Quality field with a Data_Record_Field_Name with a value of “zip”, and a Mapped_Name with a value of “MY_ZIP”, the migration utility performs a full word match to “zip” in the Python expression and replace it with “MY_ZIP”.

Data Quality Python expression:

```
record.SetField(u"zip", MY_ZIP)
```

Data Services Python expression:

```
record.SetField(u"MY_ZIP", MY_ZIP)
```

If the Mapped_Record_Field_Name is empty in the Data Quality transform, the value of the Data_Record_Field_Name is copied to the Mapped_Name and the Python expression will not contain the Data_Record_Field_Name.

New behavior for AddRecord()

The AddRecord() method behaves differently in Data Services than in Data Quality so that every time you call AddRecord(), you must call NewDataRecord().

Previously in Data Quality, when you created new records, you would call NewDataRecord() only once. When you used a “for” loop, you had to call AddRecord() over and over again. Now in Data Services, you just call AddRecord() within the “for” loop before you call NewDataRecord() .

In Data Quality, AddRecord() performed a deep copy of the object returned by NewDataRecord(). The collection made a copy for itself. In Data Services, the collection takes ownership of the NewDataRecord() object. You don't own the NewDataRecord() object anymore; instead you have a reference to it.

Cleaning up memory

Previously in Data Quality, you had to call NewDataRecord() only once to add records, and then would have to delete the object with the DeleteDataRecord() method. You had to manage memory allocated by NewDataRecord() even if you added it to a collection.

Now if you add it to a collection you don't have to manage the memory allocated by NewDataRecord(). After you create a NewDataRecord() and add it to a collection, the collection takes responsibility for cleaning up the memory.

Pre- and post-expressions in Python

After migration, the Python expression remains in the Expression option; however, the content of the expression is modified. In Data Quality, pre- and post-expressions were added behind the scenes at runtime. In Data Services, the pre- and post-expressions need to be an explicit part of the expression and are added during migration.

Multiple pipe support

Support for multiple pipes is only required for Data Quality Filter and User-Defined transform types. If the transform has more than one pipe specified in the Output_Pipes option group, or if the Python expression uses SendToPipe(), the following is performed during migration:

- Adds the Pipe_Name field to the Data_Record_Field_Name and Mapped_Name options in the Output_Fields option group.
- Adds the Pipe_Name and Data_Length fields to the Userdefined_Output_Fields option group.
- Adds a Case transform after the User-Defined transform. For each pipe specified in the Output_Pipes option group, add an output pipe to the Case transform.

After migration, you must manually set the expression on the Case transform for each pipe. You must also clean up the Python expression to remove the conditions that route records and remove the deprecated SendToPipe() calls.

Example:

For example, suppose the Data Quality User-Defined transform has two output pipes, called Passthrough and Other.

In the Data Services User-Defined transform, the migration utility adds output fields and valid output mapping. It adds a special field called Pipe_Name, which contains the value (Passthrough and Other) of the actual pipe for each of the records when the Python expression is executed at runtime.

The migration utility also adds a Data Services Case transform that uses the output field and the corresponding values in its conditional expression. The Case transform has two labels, each with the expression comparing the value of the field Pipe_Name to the actual pipe name in the User-Defined settings.

Label: Case_Pipe_PASSTHROUGH

Expression: PIPE_NAME = 'PASSTHROUGH'

Label: Case_Pipe_OTHER_PIPE

Expression: PIPE_NAME = 'OTHER_PIPE'

Same Data_Record_Field_Name on input and output

If you have a Data Quality project that contains a User-Defined transform that has input fields and output fields with the same Data_Record_Field_Name (that is, it has the same Data_Record_Field_Name on input and output, but a different MAPPED_NAME on input and output), the migration utility does not migrate the passthrough or modified fields correctly. You must correct the problem manually.

In the Python expression, the migration utility replaces the Data_Record_Field_Names with the MAPPED_NAME on input. It also adds extra lines in the pre- and post-expression code.

After migration you must manually modify the Python expression in Data Services by replacing each instance of output MAPPED_NAME with input MAPPED_NAME. An error is logged in the migration report to notify you when this manual replacement is necessary.

Example:

A field input.fld1 is used on input with MAPPED_NAME FLD_IN. The same is used on output with MAPPED_NAME FLD_OUT. You should replace FLD_OUT with FLD_IN throughout the Python expression.

Substitution variables

During migration, brackets are added around the variable value; for example `$$PATH` is changed to `[$$PATH]`.

In Data Services, substitution variables are called substitution parameters.

Executing per data flow scripts using Data Services scripting language

In Data Quality, if you had a data flow that contained scripts that must be executed once per dataflow, you used the Per Dataflow processing mode. Per Dataflow mode is not supported in Data Services.

To execute a script once per data flow in Data Services, you can rewrite the script using the Data Services scripting language. You can use the Data Services scripting language in script objects anywhere in a job or workflow (but not in a data flow). You can connect the script objects to the data flow so that

the script is executed before the data flow starts or after the data flow successfully executes. For more information about the Data Services scripting language, see the *Reference Guide*.

Related Topics

- Migration of User-Defined transform options
- Reference Guide: Scripting Language

7.4.8 Other transforms

7.4.8.1 How Data Cleanse transforms migrate

The `dqmigration` utility migrates Data Quality Data Cleanse transforms to Data Cleanse in Data Services.

7.4.8.1.1 Migration of Data Cleanse transform options

The `dqmigration` utility migrates Data Quality Data Cleanse transforms to Data Services Data Cleanse transforms with few changes. However, some options have been reorganized:

Option	Data Quality location	Data Services location
Assign Prenames Combine Compound Names Name Order Associate Name Title Enable Presumptive Name Parsing	**Options > Standardization Options**	**Options > Standardization Options > Person**
Enable Presumptive Firm Parsing	**Options > Standardization Options**	**Options > Standardization Options > Firm**
Remove Punctuation North American Phone Delimiter North American Phone Delimiter After Area North American Phone Parens Area Phone Extension Text SSN Delimiter Capitalization	**Options > Standardization Options**	**Options > Standardization Options > Other**

7.4.8.2 How Search and Replace transforms migrate

The `dqmigration` utility migrates Data Quality Search and Replace transforms to Query transforms containing the search_replace function in Data Services.

Note:
If both internal and external Search and Replace tables are used in the original Data Quality Search and Replace transform, the `dqmigration` utility generates two query transforms—one with a search_replace function using the internal table and one with a search_replace function using the external table.

Related Topics

- Reference Guide: Functions, search_replace

7.4.8.2.1 Migration of Search and Replace transform options

The following table shows how `dqmigration` utility maps the Search and Replace transform Search options to the Data Services objects and options.

Table 7-40: Search options

Data Quality Search and Replace option	Description	Data Services search_replace parameter
Case_Sensitive	Turns the case-sensitivity option on or off.	*case_sensitivity*
Default_Return_Value	Specifies a default replacement value.	*default_replace_value*
Search_Type	Specifies the type of matching to perform.	*sr_type*

Table 7-41: Search_Entry

Data Quality Search and Replace option	Description	Data Services search_replace SET options parameter
Replace_Value	Specifies the value to replace the search value.	<Replace>*replace_value*</Replace>
Search_Value	Specifies the value to search for.	<Search>*search_value*</Search>

Table 7-42: Connection_Options

Data Quality Search and Replace option	Description	Data Services object and option
Data_Source_Name	Specifies the name and path to the data source. Can also be a data source name (DSN), transparent network substrate (TNS), a connection string (for relational databases), or the actual database name.	Datastore **Data source**
Driver_Location	Specifies the path and name of the database driver.	Ignored

Data Quality Search and Replace option	Description	Data Services object and option
Driver_Name	Specifies the database driver to use when connecting from the Reader to the data source.	See table in How connections migrate
Host_Name	Specifies the server name where the data source resides.	Ignored
Password	Specifies the password appropriate to the user name in order to access the relational database for output.	Datastore **Password**
Port_Number	Specifies the port number used for connections.	Ignored
User_Name	Specifies the login name required to access the relational database for output.	Datastore **User name**

Table 7-43: Select_Options options

Data Quality Search and Replace option	Description	Data Services search_replace parameter
Replace_Value_Field_Name	Specifies the column from the returned record set that will be used for the replace value.	`replace_column`
Search_Key_Field_Name	Specifies the column from the returned record set that will be used for the search value.	`search_column`
Select_SQL_Statement	Specifies which records to select from the data source.	In SET options: `<SQLText>custom_sql_string</SQLText>`

Table 7-44: Input_Fields Field option

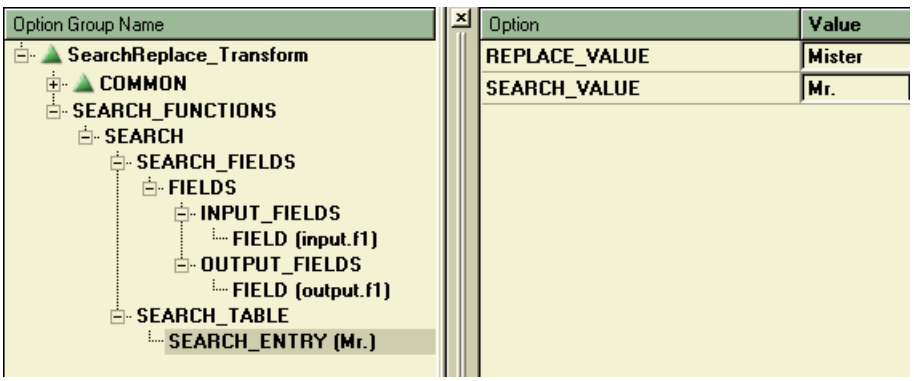
Data Quality Search and Replace option	Description	Data Services search_replace parameter
Data_Record_Field_Name	Identifies the input field that you want to process in this transform.	`input_column_list`

Table 7-45: Output _Fields Field options

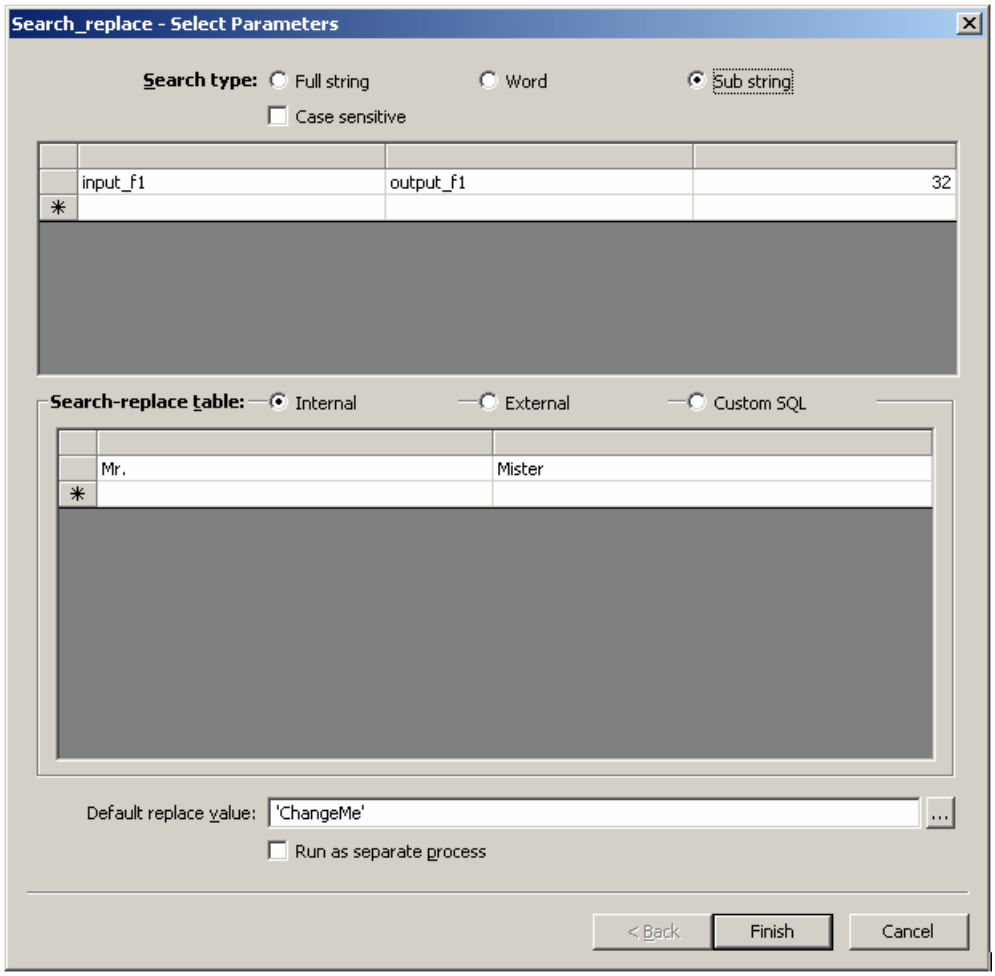
Data Quality Search and Replace option	Description	Data Services object and options
Content_Type	Specifies the type of data in the field.	Query transform output mapping Content Type.
Data_Record_Field_Length	Specifies the number of characters to make available for the output data in this field.	Query transform output mapping field length.
Data_Record_Field_Name	Specifies the name of the field.	search_replace parameters *output_column_list*, *output_variable_list*
Data_Record_Field_Type	Specifies the kind of data this output field will contain.	Query transform output mapping field type.
Input_Data_Record_Field_Name	Identifies the input field that you want to process in this transform.	search_replace paramter *input_column_list*

7.4.8.2.2 Example of migrated Search and Replace transform

This example shows how the `dqmigration` utility migrates a Search and Replace transform with an internal search and replace table.



The Search and Replace transform is migrated to a Query transform containing a search_replace function call with the following parameters.



7.4.8.3 How Global Suggestion Lists transforms migrate

The `dqmigration` utility migrates Data Quality Global Suggestion Lists transforms to Global Suggestion Lists transforms in Data Services.

The Input fields, Options, and Engines option groups remain unchanged between Data Quality and Data Services.

Related Topics

- Output field changes
- Option group changes
- Example of Global Suggestion Lists migration

7.4.8.3.1 Output field changes

The following Data Quality fields have new names in Data Services.

Data Quality field names	Data Services field names
Pick_Count	Suggestion_Count
Pick_Type	Suggestion_Type
Country	Country_Name

Data Quality GSL option	Description	Data Services GSL attribute
Content_Type	Specifies the type of data in the field.	Schema Out Content Type
Data_Record_Field_Length	Specifies the number of characters to make available for the posted data in this field.	Schema Out Type
Data_Record_Field_Type	Specifies the kind of data this output field will contain.	Schema Out Type

If you would like to change these values, select the field in the Schema Out section and make changes as necessary.

7.4.8.3.2 Option group changes

The following Data Quality option groups have new names in Data Services.

Data Quality option group names	Data Services option group names
Pick_Lists_Output_Fields	Suggestion Lists
Pick_Locality	Lastline Components
Pick_Primary_Name	Primary Name Components
Pick_Address	Address Components

In Data Quality, you could have multiple pick lists. For example, you could have Pick_Locality repeated multiple times. However, in Data Services, you can have only one Suggestion Lists option group, for example, you can have only one Lastline Components option group.

The option groups have some additional options in them.

Data Services option	Description
City Addition	Returns unofficial city information that is associated with the locality. For example, Hollywood is a vanity name for a portion of Los Angeles. "Hollywood" would be placed in the City Addition field. Suggestion List group: Available in Lastline Components and Primary Name Components options.
Locality3	The city, town, or suburb. Additional locality information goes into Locality 3. Suggestion List group: Available in the Lastline Components and Primary Name Components options.
Multiline5-6	Two additional multiline fields for individual formatted address lines. Suggestion List group: Available in the Address Components option.
Primary Name2	The street name description, for example, Primary Name1 may return "Marina" and Primary Name2 may return "The Slipway." Suggestion List group: Available in the Primary Name Components option.

You may notice that the Input/Options/Output tabs list a filter where you can sort on Best practice, In use, and All options. For migrated projects, any fields or options that were set in the original project are considered both Best practice and In use.

Note:
Singapore output data is not supported in this version of Global Suggestion Lists.

7.4.8.3.3 Example of Global Suggestion Lists migration

The following information describes some migration scenarios and how the Global Suggestion Lists transform will migrate.

Data Quality project using a pre-configured transform

Let's say that you used a pre-configured Data Quality Global Suggestion Lists base transform in your Data Quality project and you made changes to the transform options at the project level. When the project is migrated to Data Services, the data flow will now use the Data Services equivalent base transform configuration. Any options that were changed at the Data Quality project-level will also be changed and set accordingly in the data flow level in Data Services.

Data Quality project using a custom transform

Now let's say that you created a Global Suggestion Lists transform that you named my_gsl.xml and used in several projects. When the project is migrated to Data Services, the underlying transform that you created, my_gsl.xml, is migrated with all the options set as you configured them. The new file name will be dqm_my_gsl.xml. If you changed the configuration at the project level, the overrides will carry over to the dataflow level in Data Services.

We recommend that you migrate your entire repository at one time. However, if you choose to migrate on a project-by-project basis and you have a custom Global Suggestion Lists transform that is used in several projects (in the above example it would be the my_gsl.xml transform), you should migrate the custom Global Suggestion Lists transform first, and then migrate any projects that use the custom transform.

7.4.8.3.4 Web Service and Global Suggestion Lists changes

If you use wildcard searches in your web service, you can use the following wildcard characters for searches in Locality and Primary_Name fields in Data Services.

Engine	Wildcard	Example
Australia, Canada, and USA	*	Use the wildcard to complete the end of the Locality or Primary_Name. For example, Harrisb*
Multi Country	* and ?	Use the wildcard anywhere in the Locality or Primary_name. For example, Str*burg or ?view or har*.

Performance

When using wildcards, the more information you can provide, the faster the performance. For example, looking for "N*" will take much longer than if you look for "North*." Only the first 200 returns will be shown and the web service may time out during the process.

7.4.8.4 How Sorter transforms migrate

The `dqmigration` utility migrates a Data Quality Sorter transform to the Order By option of the Query transform in Data Services. When sorting partial strings of sort keys, migration creates another Query transform for temporary sort fields.

Mapping of Sorter options

All performance options in the Data Quality Sorter transform are ignored by the `dqmigration` utility because Data Services has a different sort mechanism.

In some cases, the Sorter transform options become part of the Match transform.

If the **Sort_Mode** option is set to **Collection_Sort**, the `dqmigration` utility ignores it and issues a warning message in the migration report because Data Services does not support collections. Data Services will perform a full sort.

Sorting number field types

When the sort field has a **Data_Record_Field_Type** value of **Number**, the `dqmigration` utility ignores the **Starting_Position** and **Sort_Length** options, and the sort will be on the whole key field. Data Quality uses the number field type for both decimal and integer. However, Data Services distinguishes

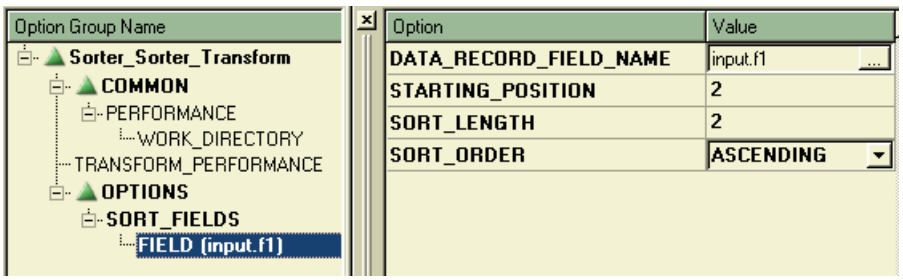
between decimal and integer data types. Therefore, migration converts a number field type to decimal (38,7) to accommodate either decimal or integer. When sorting, Data Services converts the decimal value to a character value, but this conversion adds a decimal point which distorts the sort process.

Therefore, if your Sorter transform sorts on a number field type and the value of **Starting_Position** is greater than `1`, the results will differ between the original Data Quality project and the migrated Data Services job. To correct this problem, take the following actions:

- Open the migrated data flow in the Data ServicesDesigner workspace and add another Query transform before the `Sorter_Sorter_Query`.
- Open the editor of the new Query transform and in "Schema Out", select the `Query` table name, right-click, select **New Output Column**, and select **Insert Below** on the pop-up menu.
- On the "Column Properties" window, enter a name for the substring, select an appropriate data type, and click **OK**.
- In the Mapping tab, click **Functions**, select **String Functions**, select **substr**, and click **Next**.
- In the "Define Input Parameter(s)" window, click the down arrow for **Input string**, select the previous transform name, and click **OK**.
- Select the name of the sort column and click **OK**.
- Back on the "Define Input Parameter(s)" window, enter your values for **Substring start** and **Substring length**.
- In the transform following this new Query transform, remap the new substring column from Schema In to Schema Out.

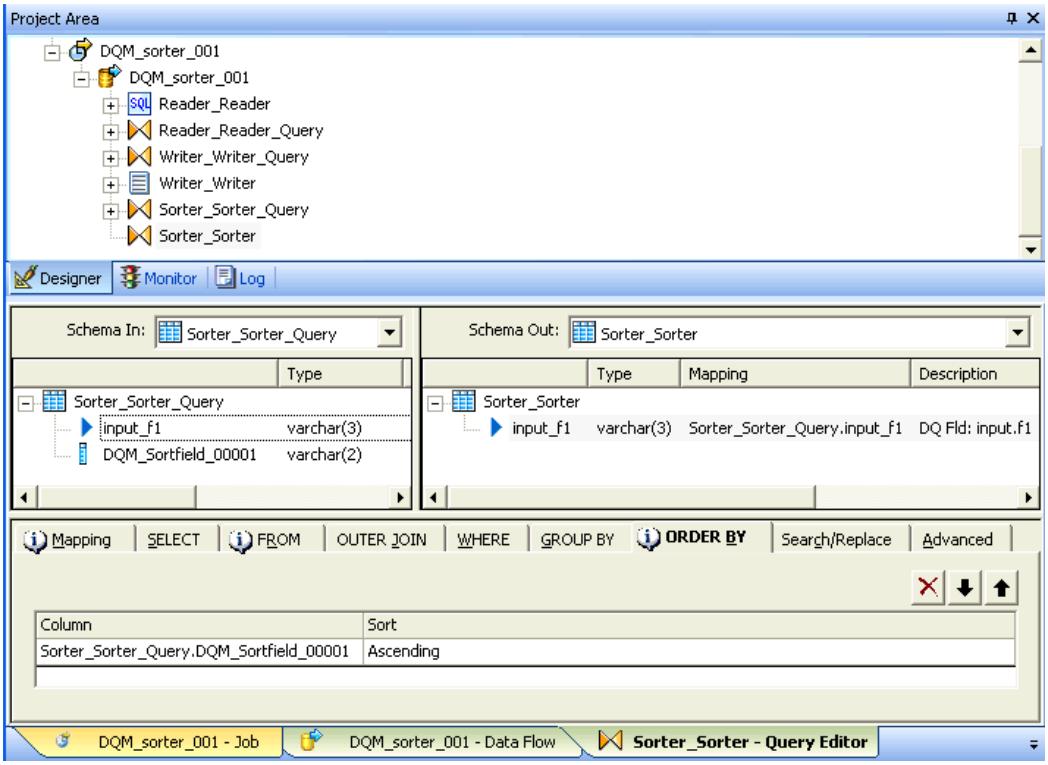
Example of Sorter transform migration

Suppose you have the following Data Quality Sorter transform. The **Starting_Position** option has the value 2 which indicates that the sort is on a partial string of the sort key.



After migration, the data flow contains two Query transforms for this Sorter example:

- The first Query, `Sorter_Sorter`, creates a temporary sort field (`DQM_Sortfield_00001`) as a substring of the sort field `input_f1`.
- The second Query, `Sorter_Sorter_Query`, performs the sort of the substring. The Order By tab in this Query transform contains the sort column and sort order, as the following Query editor shows.



Related Topics

- How Match transforms migrate

7.4.8.5 Data Quality Phonetic Key transform

The Phonetic Key transform is replaced by a Query transform that includes one of two possible functions. The first is double_metaphone() and the other is soundex().

The Double_Metaphone function encodes the input string using the Double Metaphone algorithm, and then returns a string.

The Soundex function encodes the input string using the Soundex algorithm, and then returns a string. Use this function when you want to push-down to the database-level for processing. The results may vary when you push-down to different database types.

Only use these functions for input strings in English. Any non-English characters are ignored.

7.5 Post-migration tasks

Post-migration tasks include the following:

- Improving performance
- Further cleanup
- Troubleshooting

7.5.1 Further cleanup

Most projects that now successfully execute in Data Quality will run successfully as Data Services jobs after you migrate them and fix the errors and warnings in the migration report. Open the job in Data Services and save it before attempting to run the job again.

Some migrated jobs might require some clean-up tasks before they can run successfully in Data Services. These clean-up tasks include the following:

- Fixing invalid datastores
- How to set up Group Forming section for Match
- Set up for Multiple pipe support
- Setting substitution parameter configuration
- Setting up mapped drives
- How Sorter transforms migrate

7.5.1.1 Fixing invalid datastores

Situations where the `dqmigration` utility cannot migrate a Data Quality data source to a Data Services datastore include the following:

- The data source is a MySQL database
- The Data_Source_Name is a connection string

1. Look in the Migration Report to determine whether a migrated datastore is invalid.

   For example, the Migration Details for projects\db_reader_db_writer.xml might display the following warning and error messages for datastore DQM_db_reader_db_writer_My_DB_Reader_Reader.

Job: DQM_db_reader_db_writer
DQ Source File: \projects\db_reader_db_writer.xml
Date: 02/21/2008

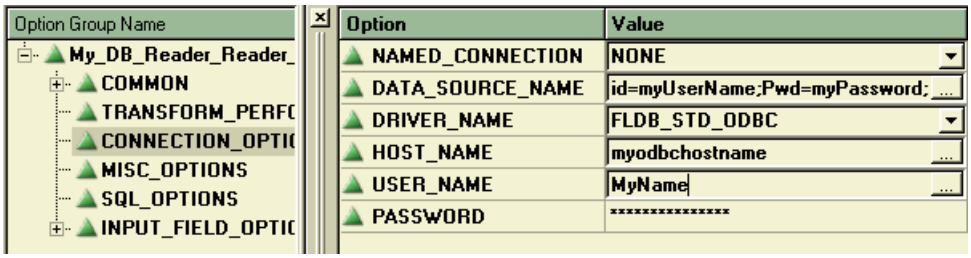
Error Warning Information

Object	Msg Type	Message	Time
MigrateFile	Info	Attempting to process file: \projects\db_reader_db_writer.xml	22:04:40.421
DQM_db_reader_db_writer_My_DB_Reader_Reader	Error	Usage of Connection string not supported	22:04:40.431
DQM_db_reader_db_writer_My_DB_Reader_Reader	Warning	Hostname was specified but will be ignored: myodbchostname	22:04:40.431
My_DB_Reader_Reader	Warning	Could not process datastore DQM_db_reader_db_writer_My_DB_Reader_Reader	22:04:40.431
MigrateFile	Info	Processing completed successfully for file: \projects\db_reader_db_writer.xml	22:04:42.945

The error `Usage of Connection string is not supported` indicates that a string was specified in the Data Quality connection options. You will need to split the string into the appropriate options in the Data Services datastore.

2. In the Data Quality Project Architect, look at the connection options to obtain the string.

   For example, the DQM_db_reader_db_writer_My_DB_Reader_Reader datastore was migrated from the following Data Quality project db_reader_db_writer.



   The value of **Data_Source_Name** for this unnamed connection is the following string:

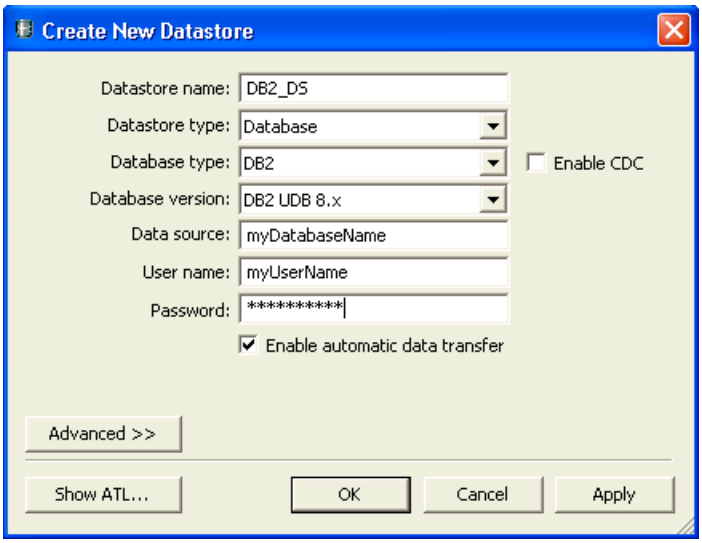
```
DRIVER={IBM DB2 ODBC Driver};Database=myDatabaseName;Hostname=myServerName;
Port=myPortNum;Protocol=TCPIP; Uid=myUserName;Pwd=myPassword;
```

3. Log in to the Data ServicesDesigner to look at the value of **Data source** in the migrated datastore. The migrated datastore is invalid if the **Data source** value has the prefix `ChangeMe`.
   a. Go to the Datastore tab in the local object library.
   b. Select the name of the datastore, right-click, and select **Edit**.

   For example, edit the datastore DQM_db_reader_db_writer_My_DB_Reader_Reader that is in the above warning message. The **Data source** value is `ChangeMeDRIVER={IBM DB2` which indicates that this datastore is not valid.

4. Create a new datastore for this data source.
   a. Obtain the connection information from Data Quality.
   b. In the Data ServicesDesigner, in the Datastores tab of the object library, right-click in the white space, and select **New**.
   c. Type a name for the datastore and enter the connection information.

      For example, use the values for Database, Uid, and Pwd from **Data_Source_Name** in Data Quality.



d. Click **OK**.

The new datastore appears in the object library.

5. Import the table's metadata from the new datastore.
   a. In the Datastores tab of the local object library, double-click the name of the new datastore.

      The items available to import through the datastore appear in the workspace.
   b. Select the items for which you want to import metadata.

      For example, to import a table, you must select a table rather than a folder that contains tables.
   c. Right-click and choose **Import**.
   d. In the Datastores tab of the local object library, expand the objects under the datastore name to display the list of imported tables.
6. Replace both the SQL transform and Query transform (that now represents the Data Quality Reader transform) with the imported table.
   a. Open the data flow for the migrated project.
   b. In the Datastores tab of the local object library, select the source table, drag it to the workspace, and select **Make Source** when the pop-up menu appears.
   c. Select the SQL transform and Query transform that represents the Data Quality Reader transform, right-click, and choose **Delete**.
   d. Connect the source table to the next transform and map the columns from the source to the transform.
7. Delete the invalid datastore.
   a. In the Datastores tab of the local object library, select the invalid datastore.
   b. Right-click and choose **Delete**.

Related Topics

- Migration report

7.5.1.2 How to set up Group Forming section for Match

You need to setup a Group Forming section for the Match or User-Defined transform in Data Services when the Data Quality Migration Report displays error messages such as the following:

```
 The User Defined Configuration %1 uses Per Collection
mode. A placeholder Group Forming section is set and requires
changes.

The User Defined Transform %1 uses Per Collection mode. A
placeholder Group Forming section is set and requires changes.

Failed to set break key for this transform. This transform requires
changes.
```

These error messages are generated when you have a per-collection User-Defined transform (or Formatter) in your dataflow, and the migration utility cannot determine what the collections are based on.

This may also occur when migration utility cannot determine the break key for a specific match transform. For example, if there is a migrated Match transform that does only post-match processing.

Here are a few examples:

1. Suppose you have a User-Defined transform (per-collection) following a Match transform in your Data Quality project. After migration, this will show up as a Match transform and a User-Defined transform. But, the Group Forming section will be incomplete in the User-Defined transform. In this case, in your Data Services data flow, you should set up the Group Forming Section of the User-Defined transform to use the Group Number field from the upstream Match transform.
2. Suppose your original Data Quality project has the following transforms: Sorter, Aggregator, and User-Defined transform (in that order), and you sorted and aggregated on the input.zip field. This would also mean that your break groups were formed based on the input.zip field. Open the User-Defined editor and set this field in the Group Forming section.
3. Suppose in your original Data Quality project, you have Match, User-Defined (per-record), and Group Statistics transforms in a dataflow (in that order). After migration, this will become Match, User-Defined, and Match transforms (in that order). The User-Defined transform is per-record, so it does not need Group Forming section. But the Match after the User-Defined transform is used for its Group Statistics post-match operation and may need the Group Forming section defined. From the data flow, you can tell that the break key for the second Match transform should be the Group Number of the first Match transform.

Related Topics

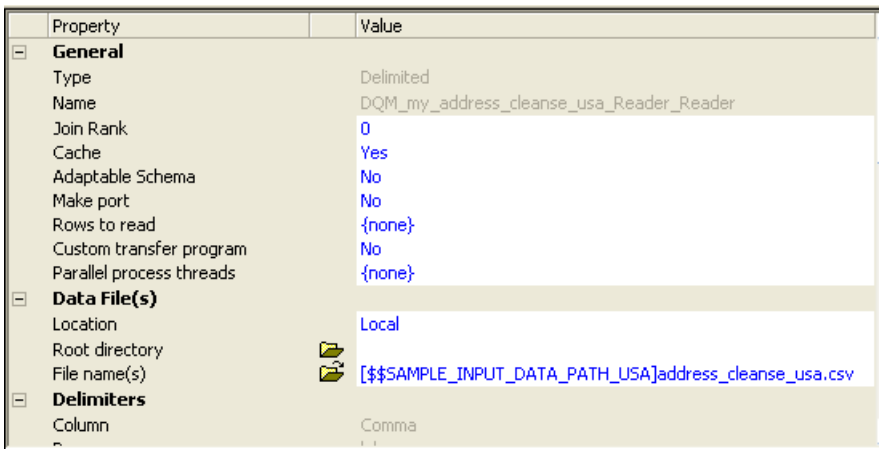
- Migration report

7.5.1.3 Setting substitution parameter configuration

When your migrated data flow contains a substitution parameter, you must take the following steps to ensure that the job runs successfully.

1. In the Data ServicesDesigner, open a data flow and open a transform that you know used a substitution variable in Data Quality and, therefore, uses a substitution parameter in Data Services.

   For example, the following source file format editor contains a substitution parameter in the **File name(s)** option.

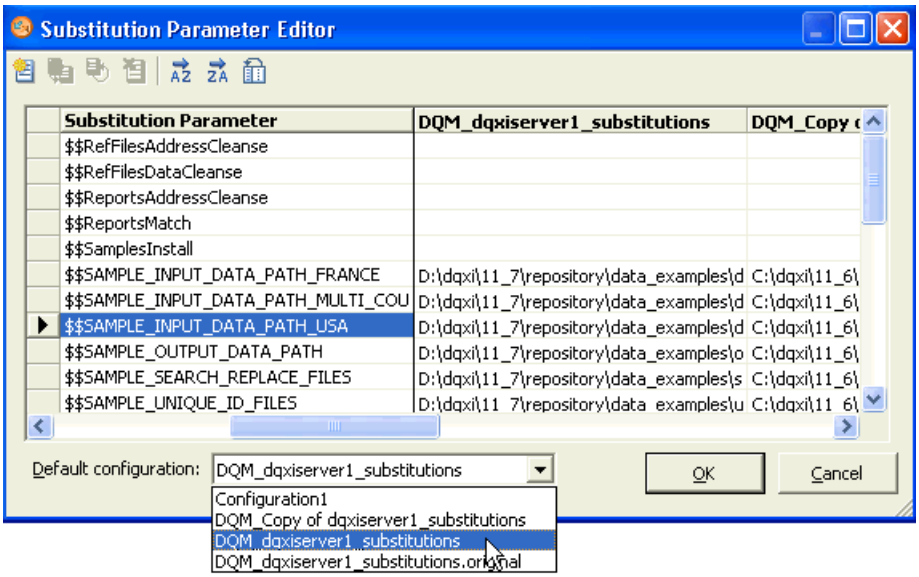


   Likewise, the target file format contains the following substitution parameter in the **File name(s)** option:

```
[$$SAMPLE_OUTPUT_DATA_PATH]
```

2. Open the Substitution Parameter Editor by clicking **Tools > Substitution Parameter Configurations...**.
3. Scroll to the right until you see values for the substitution parameters that you want to use.

   In the example, the configuration DQM_dqxiserver1_substitutions contains the values that you want to use for the substitution parameters.
4. In the **Default configuration** drop-down list, select the configuration name that you decided to use in the previous step.



5. If needed, you can change value of the substitution parameter by double-clicking the value in the configuration column.

Related Topics

- How substitution files and variables migrate

7.5.1.4 Setting up mapped drives

Data Services supports only UNC (Universal Naming Convention) paths to directories. If you set up a path to a mapped drive, the default behavior for Data Services is to convert that mapped drive to its UNC equivalent.

To make sure that your mapped drive is not converted back to the UNC path, you must add your drive names to a Job Server option in the Options window of the Data ServicesDesigner.

1. Select **Tools** > **Options**.
2. In the Options window, expand **Job Server** and then select **General**.
3. In the **Section** edit box, enter MappedNetworkDrives.
4. In the **Key** edit box, enter LocalDrive1 to map to a local drive or RemoteDrive1 to map to a remote drive.
5. In the **Value** edit box, enter a drive letter, such as M:\ for a local drive or `\\machine_name\share_name` for a remote drive.
6. Click **OK** to close the window.

If you want to add another mapped drive, you need to close the Options window and re-enter. Be sure that each entry in the **Key** edit box is a unique name.

7.5.2 Improving performance

The `dqmigration` utility migrates your Data Quality projects to equivalent Data Services data flows, datastores, and file formats so that you can run the data flows with minimal post-migration changes. However, the resulting Data Services data flows might not be optimized. The following table provides suggestions to:

- improve performance.
- facilitate future maintenance of the data flows.

Data Quality object	Data Services object after migration	Performance improvement suggestion
Named connection	Datastore	After migration, multiple datastores (one for each reader and each writer in each migrated project) might exist with the same connection information. To consolidate datastores: • Select one of the similar datastores to reuse in multiple data flows. • Find the data flows that use the similar datastores by selecting each datastore in the Local Object Library, right-click, and select the **View Where Used** option. • Replace all occurrences of the similar datastore with the chosen datastore. • Delete the redundant datastores.
Num_Of_Threads option in transforms	Not migrated	The `dqmigration` utility ignores the Num_Of_Threads option. However, it is similar to Degree of Parallelism (DOP). The effects of DOP depends on the number of processors you have available. What was suitable in Data Quality might be inefficient in Data Services. Take the following actions: • Study the new data flow. • Implement other applicable performance improvement suggestions. • Test the dataflow and then decide if you want to change the DOP setting (**Degree of Parallelism** option in the data flow Properties window).

Data Quality object	Data Services object after migration	Performance improvement suggestion
Input and Output fields	Columns in Schema In and Schema Out areas of Data Services transforms	Some input or output fields might not be used in the data flow. Delete the unused columns in the Schema In or Schema Out areas that do not need them.
Data Quality transforms	Redundant Data Services transforms or operations	These redundant transforms were part of a compound transform in the original Data Quality project. They might not be needed in the Data Services data flow. Inspect the new data flow, verify that each transform is needed, delete the unneeded ones, and remap the Schema In and Schema Out columns in the transform before and after the deleted transform. For example, you might have a compound transform that consists of a Formatter, Sorter, Aggregator, and Match transforms. This compound transform migrates to a Formatter User-Defined transform and Match transform. • If the Formatter transform was just a pass-through of fields where no formatting is really needed any longer, you can delete the Formatter User-Defined transform and directly use the fields inside the Match transform. • If the Formatter transform was used to create a field for something other than is used in the break groups, you would want to retain that transform. As another example, suppose you have a compound transform that consists of Match and Group Statistics transforms. This compound transform migrates to Match transform with a post-match group statistics operation. If your data flow does not need the group statistics operation, you should remove it.
Disabled Data Quality transforms	Active Data Services transforms	With the originally disabled transforms now active, the Data Services data flow might produce different results, as well as affect performance. If not needed, delete these originally disabled transforms from the Data Services data flow, and remap the Schema In and Schema Out columns in the transform before and after the deleted transform.

Data Quality object	Data Services object after migration	Performance improvement suggestion
Single Data Quality transform	Multiple Data Services transforms	The `dqmigration` utility adds Data Services transforms to simulate the Data Quality project. These additional transforms might not serve any purpose other than mapping fields to Schema In or Schema Out in subsequent transforms. Inspect the new data flow, verify that each transform is needed, and replace multiple transforms with a single transform. For example, a database Reader migrates to a Data Services SQL transform and Query transform. Replace the SQL and Query transforms with a single database Source and ensure that the column names of the downstream transforms use the names in the Source.
Candidate Selector	Part of new Match transform	• Consolidate datastores. • If possible, replace the custom SQL and Query transform with the source table in a datastore or source file in a file format. • If applicable, use the Join option in the Query transform.
Filter	New User-Defined transform with Python expression, followed by Case transform	If the Python expression is simplistic and has equivalent Data Services functions, change the User-Defined transform to a Case transform.
Formatter, Source, UDT	New User-Defined transform with Python expression	If the Python expression is simplistic, change the User-Defined transform to a Query transform.
Scan split	New User-Defined transform with Python expression	Query transform with replace_substr function
Search replace	New Data Services built-in function in Query transform	• Consolidate datastores. • If possible, replace the custom SQL and Query transform with the source table in the pertinent datastore or source file in the pertinent file format. • If applicable, use the decode or lookup_ext function in the Query transform.

Data Quality object	Data Services object after migration	Performance improvement suggestion
Match Family	New Match transform	After migration, multiple Match transforms might exist because of the compound Match transform settings in Data Quality. You might be to edit the data flow to use only one Match transform. Evaluate the purpose of each Match transform in the data flow and, if possible, edit one Match transform to consolidate all operations and delete the others.
Flat file Reader	Flat file source	After migration, multiple file formats (one for each Reader and Writer in each migrated project) might exist with the same connection information. To consolidate file formats, see Performance improvement suggestions for flat file Reader and Writer transforms.
Database Reader	SQL transform followed by Query transform that maps the input fields to the next transform	Replace the SQL and Query transform with the source table in the pertinent datastore.
Flat file Writer	Flat file target	Consolidate and reuse datastores. See Performance improvement suggestions for flat file Reader and Writer transforms.
Database Writer	Database target with SQL statements in Load Triggers. This target is preceded by a Query transform that maps the fields from the previous transform to the target.	Replace the migrated Query and target with the actual target table in the migrated datastore, and remap the Schema Out columns of the previous transform to the target table.

7.5.3 Troubleshooting

This topic lists the most likely migration error messages and the actions that you can take to resolve them. It also includes some warning messages for which you might need to take additional actions to make the resulting data flow run successfully in Data Services.

This topic groups the migration error and warning messages into the following categories:

- Connection errors and warnings
- Transform errors

- Other known issues

Note:
For messages that indicate "Internal error", contact Customer Support.

The `dqmigration` utility does not migrate the bootstrap and shared options files. These files are used by the Data Quality Server and transforms, and they are not applicable to Data Services. However, the migration report displays errors about these files that indicate missing attributes or unsupported file types. You can ignore these error messages.

Connection errors and warnings

Migration message	Resolution
Could not find information for *name* named connection. A default Oracle datastore will be created and used.	For each of these messages, take one of the following actions: • Create a valid named connection in Data Quality and rerun the `dqmigration` utility on that specific named connections .xml file. • Create a new Data Services datastore or file format that corresponds to the original named connection and use that in the data flow instead of the placeholder delimited file or placeholder Oracle data store that presently exists.
Could not find information for *name* named connection. A placeholder delimited file format will be created and may require changes.	
The named connection specified in the transform *name* was not found. A placeholder Oracle datastore will be created and used.	
The MySQL datastore *name* is not properly migrated and will need changes.	
The database type specified in the transform *name* is not supported. A placeholder Oracle datastore will be created and used.	Because the database or file type is not supported in Data Services, convert the data into one of the supported types, create a corresponding Data Services datastore or file format, and use that in the data flow.
File type *name* is not supported.	
Error: Usage of Connection string not supported. Warning: Could not process datastore *name*	Modify the datastore with correct connection information. See Fixing invalid datastores.
Substitution variables in connection options are not supported.	Modify the datastore with correct connection information. For example, suppose the substitution variable $$MY_USER_NAME was specified in User_Name. After migration, you must edit the datastore in Data Services and specify the actual user name.

Migration message	Resolution
Error: Could not open "*file_pathname*" format file: No such file or directory Warning: Failed to read the FMT/DMT file *file_pathname*. A placeholder delimited file format will be created but may require changes.	The format of Data Quality flat files are defined in either a delimited file format (with .dmt suffix) or a fixed file format (with .fmt suffix). When the format file is not found, the migration utility creates the format based on the input/output field settings which are defined in the reader/writer in Data Quality. These settings might not match the actual file layout and will cause incorrect processing results. Ensure that the .dmt or .fmt file format is suitable for the actual data to process.
Failed to read DATA_SOURCE_NAME for *name*. A placeholder delimited file format will be created and may require changes.	
Database type *name* is not supported in this version. A placeholder Oracle datastore will be created and used.	Because the database type is no longer supported, convert the data into one of the supported types. Create a corresponding datastore or file format and use that in the data flow.
The MySQL datastore *name* is not properly migrated and will need changes.	This error occurs for a MySQL database with a Server Name connection option. Create an ODBC Data Source and edit the placeholder MySQL datastore to correct the connection information to the ODBC Data Source.

Transform errors

Migration message	Resolution
Could not process datastore *name* for Candidate Selection section.	Create a new datastore as required and use that in the Candidate Selection section. See the Match transform in the Reference Guide.
DQ Candidate Selector option ALL_RECORDS_IN_COLLECTION is not supported. The migrated behavior is always as FIRST_RECORD_IN_COLLECTION.	Data collections do not exist in Data Services. The SELECT statement will run on the first record only.
DQ option TRANSFORM_PERFORMANCE/ BUFFER_OVERFLOW_OPERATION is not supported anymore. The value of MAX_DATA_COLLECTION_SIZE is set to 0.	Because data collections do not exist in Data Services, the values in the TRANSFORM_PERFORMANCE option group are not used for MAX_DATA_COLLECTION_SIZE in the Misc_Options group. A value of 0 for MAX_DATA_COLLECTION_SIZE catches most duplicates because Data Services uses a pageable cache if the size of your data exceeds two gigabytes of memory.

Migration message	Resolution
A duplicate field name was detected: *transformname.fieldname*	Ensure that the duplicate field name is needed. If it is needed, remap it in your transforms. If it is not needed, delete it.
Failed to assemble document *name1* after running RUN_ON_NOT_ASSEMBLED scripts. Error: The file *name2*, which is referenced by the file *name3*, could not be found.	Ensure that the second specified file exists in the source Data Quality Repository.
Failed to set break key for this transform. This transform requires changes.	Determine the break key to use in the Group Forming section of the Match transform in Data Services data flow. For details, see How to set up Group Forming section for Match
An input field *name* is specified for transform *name2* but it is not present as an output field in any of the upstream transforms.	Take one of the following actions: • In the original Data Quality project, ensure that the field is actually present and the name is spelled correctly, and rerun the `dqmigration` utility. • If the field is not needed, delete it from the data flow in the Data ServicesDesigner.
Output Length for field *name* has been changed from *length1* to *length2*. Data truncation may occur. Please adjust output schemas.	This error is applicable to the USA_Regulatory_Address_Cleanse, Global_Address_Cleanse, and Global_Suggestion_List transforms. Check subsequent transforms downstream to ensure that the increase in length has not caused truncation. If truncation has occurred, increase the length in the applicable downstream transforms.
Project *name* has been skipped during migration possibly due to earlier migration errors.	Fix the errors in the original Data Quality project and rerun the `dqmigration` utility.
Transform *name* has been skipped during migration possibly due to earlier migration errors.	Take one of the following actions: • If the transform is supported in Data Services, fix the errors in the original Data Quality project, and rerun the `dqmigration` utility on that specific project. • If the transform is deprecated or not useful and it was in the middle of the data flow, delete the transform in the Data Services data flow, and make appropriate connections between the preceding and succeeding transforms to each other.

Migration message	Resolution
The User-Defined Configuration *name* uses Per Collection mode. A placeholder Group Forming section is set and requires changes.	Determine the break key to use in the Group Forming section of the Match transform in Data Services data flow. For details, see How to set up Group Forming section for Match
The User-Defined transform *name* uses Per Collection mode. A placeholder Group Forming section is set and requires changes.	
Transform *name* is disabled in the Data Quality project, but it is migrated as an active transform.	If the transform in question does not adversely affect the results, then delete the transform and reconnect the preceding and succeeding transforms to each other. **Note:** Leaving redundant transforms might impact performance.
Match transform *name* option error Option Prioritization/Priority_Value/Field/Blank_Penalty: At least one Blank_Penalty option must be present in this group.	In the migrated data flow, open the Match transform and enter 0 as a default value for the options that are blank in the Blank_Penalty option group.
Best record has a destination field that is blank.	Take one of the following actions: • Before migration, complete the Best Record Destination field. • After migration, manually remove the broken fields in Data Services
Use of periods in the Input_FIeld_Name value will not output those fields	If you have a value with a period in it, for example Family.Name, in your Input_Field_Name field, then the migration utility cannot process the field. Change the value in the Input_Field_Name to remove all periods. You may use underscores.
Duplicate fields in the case transform during validation	When you validate your migrated job in Data Services and see an error in the case transform about duplicate fields, take one of the following actions: • change the field name in the Data Quality data flow to make it unique. • change the duplicate name in the Data Services job to make it unique.

Migration message	Resolution
Transform Match:Option Error *option name* has a value of *value name* which cannot be repeated. Deselect this output field and select again so that it will be included only once.	The Match transform has output the option twice in the Output Fields section. To correct the problem: 1. Open the Match transform (NOT the Match Editor). 2. Write down the group number output fields, if you want to recreate them in step 7. 3. In the Output Schema, right-click on all but one of the group number output fields and select Delete. 4. Close the Match transform editor. 5. In the dataflow, create a new Query transform and connect it to the right of the Match transform. 6. Open the Query transform and map the input and output fields. 7. Create the same number of new output fields as you deleted in the Match transform with the Type equal to Group Number. 8. Map the group number input fields to the new group number output fields. 9. Save and run the dataflow.

Other known issues

Global Address Cleanse paths are not migrated correctly:

In the directory paths in the Global Address Cleanse transform, the reference file paths are migrated incorrectly. To correct the path, open the dataflow in Data ServicesDesigner, open the Global Address Cleanse transform, go to the Option Explorer and modify the path.

Option name	Incorrect path	Correct path
Global Directory	[$$DIR_PATH]_GLOBAL_ADDRESS	[$$DIR_PATH_GLOBAL_ADDRESS]
Australia Directory	[$$DIR_PATH]_ AUSTRALIA	[$$DIR_PATH_AUSTRALIA]
Canada Directory	[$$DIR_PATH]_ CANADA	[$$DIR_PATH_CANADA]
USA Directory	[$$DIR_PATH]_ USA	[$$DIR_PATH_USA]

Index

G

I

J

L

M

N

O

P

Q

R

S

T

U

V

W