TIBCO® Data Migrator

User's Guide

Release 8207
June 2021
DN3501987.0621

Copyright © 2021. TIBCO Software Inc. All Rights Reserved.

Contents

1. Introduction to TIBCO® Data Migrator 13

Data Management and Migration Overview 14
What Is TIBCO Data Migrator? 14
Architecture 15
TIBCO WebFOCUS Reporting Server 15
Data Management Console 17
Web Console 20
Designing Data Flows and Process Flows 21
Revised Functionality 22
Testing the TIBCO WebFOCUS Reporting Server. 22
Quick Copy. 23
Keeping the TIBCO WebFOCUS Reporting Server Connection Alive. 23
File Listener. 24
COMT and TABT Target Files. 24
Custom Reports for the Log and Statistics Tables. 24
Run Deferred (Legacy). 25
Test Run and Run. 25
Database Default Values. 25
Unassigned Fields. 25
Getting Help 26

2. Setting Up TIBCO Data Migrator 27

Setup At a Glance 28
Remote TIBCO WebFOCUS Reporting Server Setup At a Glance 30
TIBCO Data Migrator Setup 31
Initiating TIBCO WebFOCUS Reporting Server Administration Tasks 38
Authorizing TIBCO WebFOCUS Reporting Server Usage and Administration 38
Managing Application Directories and Configuring the Application Path 41
Configuring Data Adapters 48
Creating Metadata 54
Testing Sample Data and a Process Flow 64
Performing TIBCO WebFOCUS Reporting Server Administration Tasks 64
Configuring TIBCO WebFOCUS Reporting Server Operations. 65

Using Secure Shell FTP (SFTP) 78
Using SFTP From Windows. 79
Using SFTP From Early Windows Versions. 81
Upgrading TIBCO Data Migrator From Release 4.3 or 5.1 82
Upgrading TIBCO Data Migrator From Release 5.2 and Higher 82
Port Number. 83
Configuration Files. 83
Profiles and Applications. 83
Data Flows. 83
Scheduler. 83
Data Management Console. 83
Upgrading TIBCO Data Migrator from Release 7.6 85
Upgrading TIBCO Data Migrator to Release 7.7.07 or 82 86

3. Planning a TIBCO Data Migrator Application 87

Designing Your Application 87
Using Synonyms for Data Sources and Data Targets 88
Reviewing Your Data Sources 92
Accessing Data Sources. 93
Reviewing Your Data Targets 94
Relational Data Targets. 95
Non-Relational Data Targets. 96
Remote Target Destinations. 96
Establishing Security 98
Running Flows. 98
Restricting the Application Paths Available to a User. 99
Running Scheduled Flows Under a User ID. 100

4. Working in the Data Management Console 103

Data Management Console Interface 103
Application Menu. 105
Quick Access Toolbar. 106
Ribbon. 107
Ribbon Organization. 108

Ribbon Tabs. 108
Style Menu. 108
About Button. 108
Shortcut Keys for Navigating the Workspace. 109
Navigation Pane 110
Copying, Cutting, and Deleting Procedures, Synonyms, and Other Files. 122
Workspace 123
Data Flow Tab. 124
Working With Columns, Variables, and Functions. 126
Process Flow Tab. 129
Text View Tab. 130
Text Editor 131
Aerial View 133
Ribbon Tabs 133
Home Tab. 133
Flow Tab. 136
User Functions Tab. 139
Reports Tab. 140
Metadata Tab 142
Text Editor Tab 142
Console Log 142
Session Log. 143
Setting Flow Properties 144
Viewing Properties 152
Application Directory Properties. 155
Flow or Procedure Properties. 156
Setting User Preferences 156
User Privileges 167
Configuring National Language Support 169
Ending a Data Management Console Session 169

5. Designing a Data Flow 171

The Data Flow Interface 172

Simple Data Flow at a Glance 174
Complex Data Flow at a Glance 175
Creating a Data Flow 177
Adding and Deleting Objects in Your Data Flow. 178
Connecting and Disconnecting Objects in Your Data Flow. 181
Saving a Data Flow. 182
Specifying Data Sources 183
Updating the List of Data Sources. 189
Creating a Source Transformation in a Data Source Object 190
Order of Evaluation. 203
Changing the Order of Evaluation. 204
Joining Data Sources 204
Adding a SELECT to a Data Flow 218
Adding a UNION to a Data Flow 218
Selecting Columns 220
Aggregating Columns. 225
Creating an SQL Calculation. 226
Using an SQL Function. 230
Testing the SQL Statement 242
Specifying Filters in a Data Flow 245
Specifying Sorts 251
Selecting a Data Target 253
Specifying Data Target Options 261
Target Load Options. 262
Extended Bulk Load Utility Options. 266
Using Direct Bulk Load for a Flow. 279
Target Properties Pane. 289
Loading Data to Salesforce.com. 303
Mapping Transformations 305
Validating Records. 314
Using Variables in a Flow 314
Using a Parameter With a Flow. 320
Using Functions 325

Column Formats and Mapping 343
Setting Properties of a Flow 355
Running a Data Flow 356
Using the Cluster Manager Scheduler. 356
Migrating Data Flows 361
Creating a Direct Load Flow 363
Creating a DBMS SQL Flow 365
Creating a Quick Copy 367

6. Designing a Process Flow 369

Process Flow Methodology 370
Creating a Process Flow With a Data Flow. 370
Creating a Process Flow Without a Data Flow. 370
Process Flow Interface 370
Simple Process Flow At a Glance 374
Complex Process Flow at a Glance 375
Creating a Process Flow 376
Saving a Process Flow. 379
Adding a Saved Procedure to a Process Flow 380
Assigning and Using Variables in a Process Flow 382
Maintaining Continuous Processing in a Single Agent 391
Executing a Process Flow That Contains Only Procedures 393
Adding Email Messages to a Process Flow 394
Setting Connection Logic 395
Branching in a Process Flow. 401
Grouping and Controlling Simultaneous Execution of Flows 401
Running a Process Flow 405
Scheduling a Process Flow 406

7. Tutorial: Creating Data and Process Flows 415

What You Are Going To Do 416
Startup and Setup Tasks 416
Start Your TIBCO WebFOCUS Reporting Server and Data Management Console. 416
Configure a TIBCO WebFOCUS Reporting Server Node (Optional). 417

Create Sample Data and Metadata and Create an Application Directory. 419
Exercise 1: Creating the Data Flow 419
Selecting and Joining Data Sources 420
Add Your First Data Source. 421
Add and Join a Second Data Source to the First. 422
Create A Virtual Column in the Second Data Source. 424
Assign Join Properties. 426
Selecting and Managing Columns 428
Select and Create Columns. 429
Filter the Data. 433
Adding Data Targets 436
Map Columns in an Existing Data Target. 436
Create a Data Target. 441
Save and Name the Data Flow. 444
Submitting the Data Flow 444
Data Flow Review 445
Exercise 2: Creating the Process Flow 445
Starting the Process Flow 446
Adding a Data Flow Object to the Process Flow 446
Save and Name the Process Flow. 447
Branching in the Process Flow 447
Create the Procedures. 447
Connect the Procedures to the Data Flow. 448
Specify Your Connection Logic. 449
Submitting the Process Flow 449

8. Generating Reports, Viewing the Log File, and Printing Flows 451

Log, Statistics, and Summary Flow Reports 451
Scheduler Reports 456
Impact Analysis Reports on Synonyms, Procedures and Columns 459
Working With a Report 465
Viewing a Detailed Flow Report and Log File 470
Printing a Graphical Representation of a Data or Process Flow 473

9. Optimizing Performance 475
Improving Data Extraction Performance 475
Using Automatic Passthru 476
Avoiding Non-Automatic Passthru. 477
Improving Data Loading Performance 479
Improving Lookup Performance 480
Bulk Retrieval With Salesforce.com 482
10. Working With Synonyms 485
Synonym Editor Layout 485
Viewing and Editing Synonym Attributes 499
Pivoting Repeating Columns in a Synonym. 510
Using Variables in a Synonym 520
Variables Attribute Summary 522
Use a Variable in a Synonym 523
Creating and Enhancing Synonyms Using Modeling View 523
Adding Indexes to a Synonym 526
Adding Virtual Columns (DEFINE) in a Synonym 527
Viewing Data Profiling Characteristics 530
Data Profiling a Synonym or Segment. 530
Data Profiling a Single Column. 533
Viewing Data Lineage for a Column 538
11. Advanced Tips and Techniques 541
Overview of Examples 542
Creating Sample Data and Metadata 543
Horizontal Partitioning with Multi-Table Load 544
Using the Decode Function 548
Generating a Surrogate Key 551
Generating a Surrogate Key Using Slowly Changing Dimensions 557
Mapping to a Sequential File of a Specific Format 561
Creating and Using Reusable Transformations 563
Using Multiple Targets to Create a Single File With Multiple Records Types 568
Creating a Target File With Multiple RECTYPEs 571

Combining Multiple Source Rows in a Single Target Row 575
Using Lookups to Create Multi-Table Flows Without Joins 579
Record Logging 583
Using Variables in Data Flows 586
Creating a Data Flow Using a Target Based on a Predefined XML Schema 590
Loading a Target Table Using the File Listener 594
Sending a Target File to an FTP or SFTP Server 601
Loading a Star Schema 605
Iterating Through Multiple Tables 612
Using Subqueries 620
Loading Images Into a Database Table 623
Updating Values in a Table 627

12. Change Data Capture 635

Enabling the Database for Logging in Relational Databases 635
Configuring CDC in Microsoft SQL Server®. 635
Configuring CDC in MySQL. 637
Preparing the Environment for CDC with MySQL®. 637
Java®. 637
Connector/J. 638
Utilities for Processing Binary Log Files. 638
Configuring MySQL. 639
Configuring CDC in Oracle®. 639
Configuring CDC in PostgreSQL®. 641
Preparing the Environment for CDC With PostgreSQL. 641
Using Logical Decoding and Monitoring Slots. 644
Configuring the Adapter for PostgreSQL. 646
Configuring CDC in IBM Db2 Universal Database® on Windows and UNIX. 647
Configuring CDC in IBM Db2 Universal Database on System Z. 647
Configuring CDC in IBM Db2 Universal Database on iSeries. 648
Creating a Journal Library. 649
Turning a Library Into a Journaling Library. 650
Enabling Journaling on a Not Journaling Library. 651

Configuring CDC in IBM Db2 Universal Database on z/OS. 653
Configuring CDC in MariaDB®. 653
Preparing the Environment for CDC with MariaDB. 653
Java. 654
Connector/J. 654
Utilities for Processing Binary Log Files. 654
Configuring MariaDB. 655
Properties of a Table Log Record Synonym 656
Considerations for Table Log Records Synonyms. 658
Tutorial: Using Change Data Capture with Relational Databases 659

A. Log Statistics and Tables 669

Log and Statistics Tables 669
TIBCO FOCUS® Logs and Statistics 670
CSV Logs and Statistics 670
Managing Log and Statistics 671
Custom Procedures for Managing TIBCO FOCUS Logs and Statistics 673
Description of the Log Files 674
Description of the Statistics Files 675
List of Essential ETL LOG Message Codes 677

B. Event-Based Scheduling 681

Starting Flows With CMRUN 681
Specifying an External Password. 692
Starting Flows With CMASAP 693
Starting Flows From TIBCO FOCUS 695
Determining the Status of an Event-Based Flow 695

C. Problems, Errors, and Troubleshooting 697

TIBCO Data Migrator Status Codes and TIBCO WebFOCUS Reporting Server Messages 697
TIBCO WebFOCUS Reporting Server Messages. 698
TIBCO WebFOCUS Reporting Server Log Messages for Agent Termination. 700
TIBCO WebFOCUS Reporting Server Log Message for a Missed Scheduled Request. 700
Connector Communications Errors. 700
TIBCO WebFOCUS Reporting Server Run-Time Error Codes 701

TIBCO Data Migrator Management Problem Resolution 702
Including Run-Time Messages in the ETL Log 703
Performing a Trace to Diagnose TIBCO WebFOCUS Reporting Server Problems 703

D. Customizing Your Environment 705

TIBCO Data Migrator Parameters 705

E. Source Management 709

Configuring Source Management and Setting Up Source Control for Git 709
Configuring Source Management and Setting Up Source Control That Uses SCC API 725
Setup, Configure, and Associate TIBCO Data Migrator to Work With ChangeMan® DS. . . 727
Setup, Configure, and Associate TIBCO Data Migrator to Work With ClearCase®. 729
Setup, Configure, and Associate TIBCO Data Migrator to Work With Concurrent Versions System (CVS). 731
Setup, Configure, and Associate TIBCO Data Migrator With Subversion® 734
Setup, Configure, and Associate Data Migrator to Work With Team Foundation Server (TFS). 741
Source Control Security Server and Client Setup. 748
Using a Source Control System to Manage Procedures and Synonyms. 752

Legal and Third-Party Notices 759

Chapter 1

Introduction to TIBCO® Data Migrator

TIBCO Data Migrator, formerly ETL Manager, represents a broad category of tools designed to facilitate and automate the extraction and integration of data from various sources. At various points during the process, from source extraction through target load, data is *transformed* through the application of business rules. Once the transformation is complete, the data is loaded into table structures that have been optimized for a particular application.

Data Migrator automates this process.

For up-to-the-minute information, please refer to the release notes in the *Help* menu of the Web Console.

In this chapter:

- Data Management and Migration Overview
- What Is TIBCO Data Migrator?
- Architecture
- TIBCO WebFOCUS Reporting Server
- Data Management Console
- Web Console
- Designing Data Flows and Process Flows
- Revised Functionality
- Getting Help

Data Management and Migration Overview

Data Migrator management and migration is the process of extracting data from one or more data sources and moving that data into one or more data targets.

- A data source often contains operational data, such as sales, orders, customers, invoices, and inventories. The data source is designed for online transaction processing and high-speed updates. Systems that maintain these data sources are often highly volatile and mission critical. Within a company, there may be many such systems containing related information with no means for cross-structure comparison.
- A data target, referred to as the data warehouse, data mart, or another type of target, is often relational and is generally located on a machine with optimal performance and availability. This data target typically stores information historically, and is geared for high-performance online analysis.

Often, the process of ETL (Extract-Transform-Load) management involves data cleansing, such as numeric aggregation, improvements to cryptic names, or the calculation of new columns. This process yields many benefits, including:

- **Ease of use.** The data warehouse may be optimized for analysis, and the data cleansing process ensures that data is stored in a more readable format.
- **Reduced load on legacy systems.** Since your analysts are no longer accessing your operational data sources, production systems are not impacted by complex queries.
- **Better response time.** The data warehouse can be situated on a local department machine. You no longer have to search the network to find your data.
- **Ownership.** Many departmental data warehouses can be created, each one containing summary information for that department only.

What Is TIBCO Data Migrator?

Data Migrator is a suite of software components that automates the process of building and maintaining a data warehouse or data mart. Using Data Migrator, you can:

- Access source data in numerous formats and operating systems.
- Integrate multiple data sources into a single target or multiple data targets.
- Apply powerful data cleansing rules and transformation logic.
- Aggregate data and create roll-ups to aid decision support.

- Use specialized high-volume data loaders.
- Schedule data updates at user-defined intervals, triggered by events, or based on conditional dependencies.
- Load a Star Schema with Slowly Changing Dimensions.
- Monitor and manage key server functions.
- View comprehensive logging and transaction statistics.

Since Data Migrator is a true client/server middleware solution, underlying communications protocols and data source subsystems are transparent.

Architecture

Data Migrator is comprised of two basic components and one supplementary component:

- **TIBCO WebFOCUS® Reporting Server** A WebFOCUS® Reporting Server must be installed and configured to support Data Migrator operations. The server houses and executes data and process flows, which are the key DM design elements. For details about server operations, see *TIBCO WebFOCUS Reporting Server* on page 15.
- **Data Migrator Data Management Console (DMC).** The main component of the DMC is the Data Migrator graphical user interface. It runs on a Windows® PC, where you design, test, and run data and process flows. For details about elements of this console, see *Data Management Console* on page 17.
- **Server Web Console.** The Server Web Console is installed with the server. It enables you to perform browser-based server administration tasks. For a list of Data Migrator-related server administration tasks, see *Web Console* on page 20.

You must install and/or configure these components before you can begin using Data Migrator to build the data flows and process flows that will build your data targets.

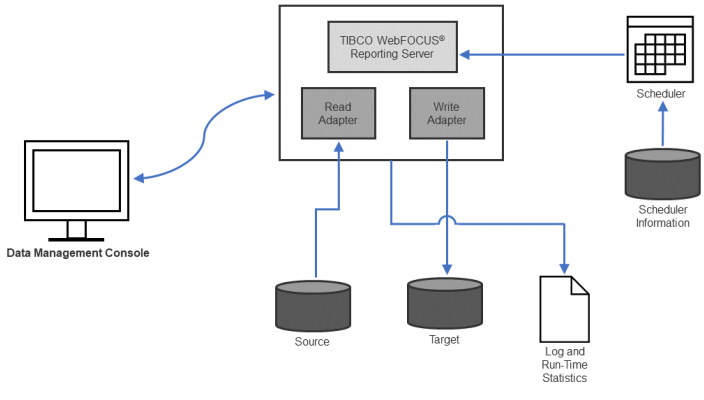
TIBCO WebFOCUS Reporting Server

A WebFOCUS Reporting Server must be installed and configured for Data Migrator. The server houses and executes the data and process flows from which a data warehouse, data mart, or other data targets are constructed. The server consists of the following components:

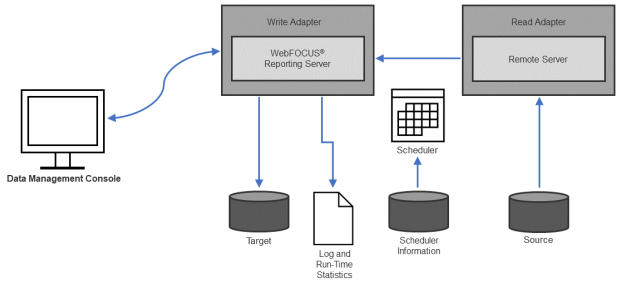
- Engines for data sorting, transformations, email notification, scheduling, and security rules.
- Read and write adapters, which are used to extract and load the data.
- Metadata or synonyms (.mas and .acx files) that describe column information and access rules to data sources and/or data targets.

- Data flows, which are procedures that contain the necessary instructions for extracting, transforming, and loading the data targets. These procedures are built by the DMC and stored on the server in two files:
  - A procedure with the extension .fex, which is ready to run.
  - A file with the extension .etg, which contains detailed information needed for the graphical display.
- Process flows, which contain the necessary instructions for interleaving data flows with other objects. Like data flows, these instructions are built by the DMC and stored on the server as .fex and .etg files.
- Data Migrator internal tables, containing the DM log and run-time statistics.
- Stored procedures, which contain instructions for specific tasks. These procedures are edited using the text editor of the DMC.
- Remote server (optional). Access to a remote server is required when source data is on a different platform from the Reporting Server. Adapters to extract the source data reside on the remote server (subserver). Though optional, one or more subservers are used at many Data Migrator sites.

The following figure illustrates a Reporting Server environment in which both the data source and the data target reside on the Server.



The following figure illustrates a Reporting Server environment in which the data target (whether a data warehouse, data mart, or another type of target) resides on the Reporting Server and the data source resides on a remote server (subserver).



Data Management Console

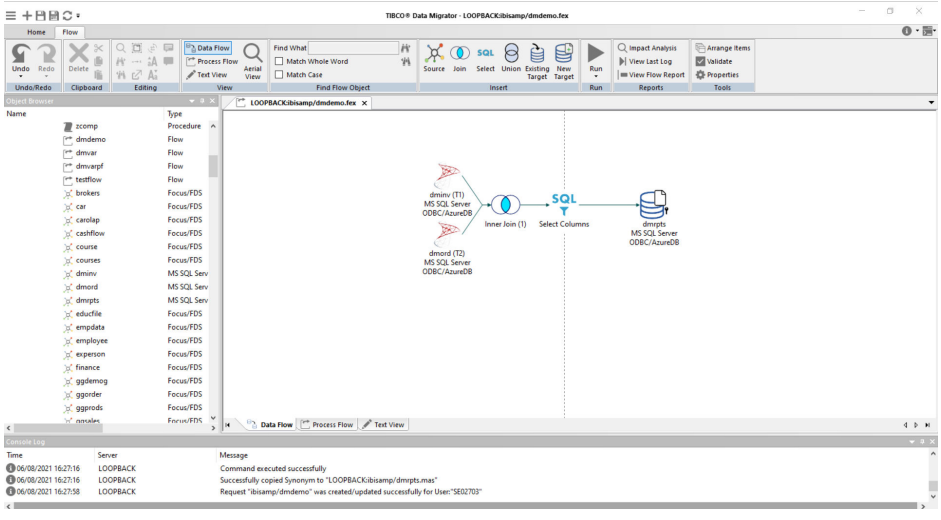
The Data Migrator Data Management Console (DMC) is a graphical design interface that runs on Windows 7, 8.2, 10, and Windows Server 2008 and 2012. You can use the DMC to:

- Design and test data flows and process flows. For more information, see *Designing Data Flows and Process Flows* on page 21.
- Create and run stored procedures.
- Add and configure data adapters.
- Test synonyms and data adapter configurations.
- Manage metadata by creating and editing synonyms.
- View log and report statistics.

Note: Ancillary functions, such as scheduling and email notification for processing, can also be performed from the Web Console, which you can launch from the DMC. For more information, see *Web Console* on page 20.

This manual focuses on the tasks you perform from the DMC. For more information on the Reporting Server and the Server Web Console, see the *TIBCO WebFOCUS® Reporting Server Administration* manual.

The following image shows the main window of the DMC in Data Flow mode.



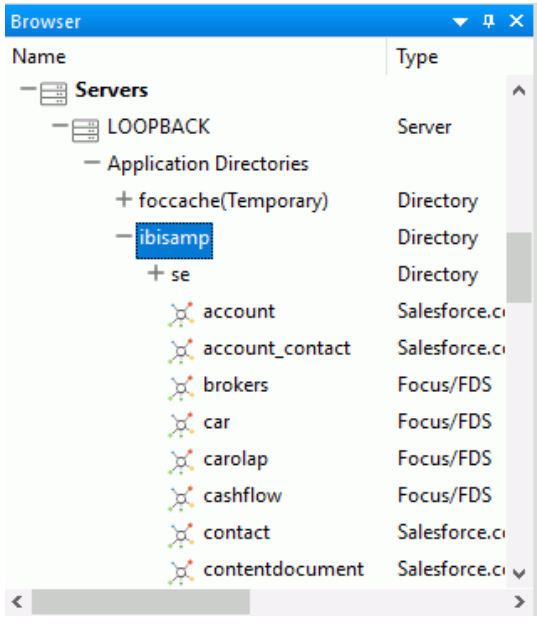
It includes the following components:

- **The Navigation pane.** On the left, always displays a list of servers that can be accessed by Data Migrator and on which you can store flows.

  Each application directory displays the files for data flows, process flows, any supporting stored procedures created for an application, and metadata.

  Note: When you are logged on to a Server, you can view the Application Directories , Adapters, and Workspace folders in this pane. Until you log onto a server, only the servers appear.

Each application directory also displays the synonyms that are available for source and target data in the data flows of an application. The synonyms in the pane shown in the following image describe data for a variety of database systems.



Note: If you do not see the synonyms, you must select the synonyms button , located in the Filter group of the Home tab.

- **The Data Flow workspace.** At the right, where the flow is designed.
- **The Ribbon.** At the top, provides buttons for the actions most commonly performed from the DMC. Tools available on the ribbon correspond to the task currently being performed. The tabs that can appear on the ribbon are:
  - A Home tab for familiar tasks, such as language selection.
  - A Flow tab for key aspects of the design tasks.
  - A Report tab for tasks associated with report output.
  - A Metadata tab for editing synonyms.
  - A Text Editor tab for editing procedures and synonyms in text view.
- **The Console Log.** Appears below the workspace and displays application and server messages.

For detailed information about these features of the DMC, see *Working in the Data Management Console* on page 103.

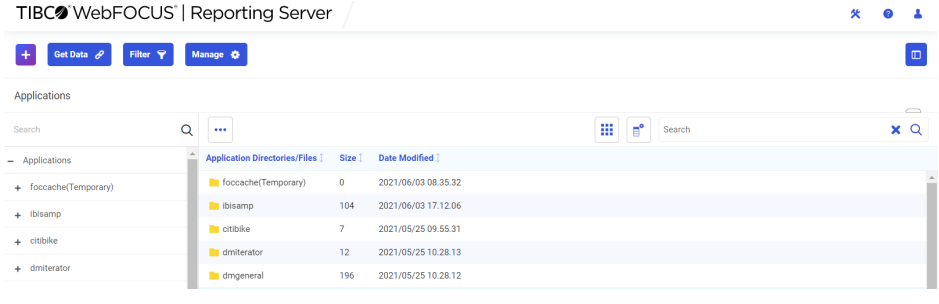
Web Console

Installed with the Reporting Server, the Web Console, accessed through a browser, can be used to perform Server administration and monitoring tasks, including the following:

- Controlling server security.
- Monitoring and administering running agents.
- Configuring email notification.
- Monitoring the execution of flows.
- Debugging data and process flows.
- Configuring the Reporting Server and remote servers.
- Accessing DM and Scheduler reports.
- Configuring data adapters.
- Creating metadata required for data extraction and loading.
- Accessing help documents and release notes.
- Creating flows.

  Note: When you design a flow in the Web Console, you cannot add new FOCUS targets.

The following image shows the Web Console, which can be accessed directly or from the DMC. From here you can open web pages on which all of these functions can be performed.



For information about key server functions required for Data Migrator, see *Initiating TIBCO WebFOCUS Reporting Server Administration Tasks* on page 38.

In addition, server administration is fully documented in the *TIBCO WebFOCUS® Reporting Server Administration* manual, which is accessible as online help from the Web Console.

Designing Data Flows and Process Flows

At the core of Data Migrator is the ability to design data flows and process flows, which are types of procedures. The Data Migrator Data Management Console facilitates the design task using two tabs, Data Flow and Process Flow, and provides a Text View tab from which you can see the code generated by the graphical tools.

- **Data Flow.** Normally, you design a data flow from a set of objects that identify your data source(s), your data target(s), the columns of data to include in the data target(s), and the business rules that govern how the data is extracted from the source(s), transformed, and loaded into the target(s).

  You can join multiple data sources, filter and sort the extracted source data, and specify column mappings from the source to the target.

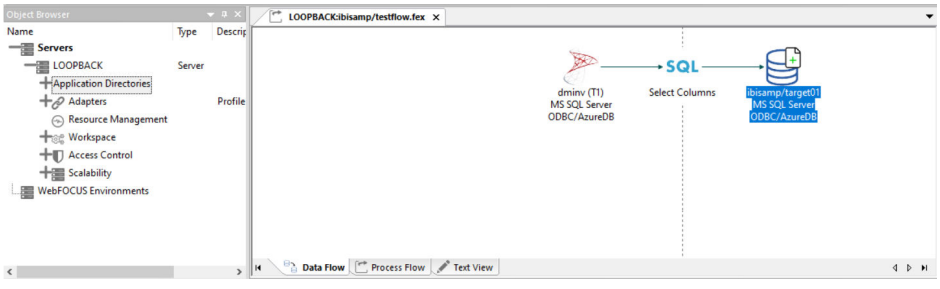
- **Process Flow.** You structure how the data flow is processed by positioning objects in your workspace and defining their interactions. You can combine:

  - A Start object on a process flow defines the destination of the log and output of the submitted process, and defines the schedule of the process.
  - Data flow objects that indicate what data to extract and move from data sources to data targets.
  - Email objects that notify users about the status of the process at specified points.
  - Stored procedure objects that perform a variety of supplementary tasks before or after the extraction and load steps defined in a data flow.
  - Variable objects that set values for the variables used in a flow.
  - Connector objects, represented as arrows, that specify execution logic for the objects in the process flow (for example, what to do next when a step succeeds or fails).
  - Group objects that specify parallel, rather than sequential, execution of a subset of objects within the total flow. Objects that follow are not executed until processing for the grouped objects is complete.

- **Text View.** This tab enables you to toggle between the graphical interface and the underlying code at any point in your design process.

Example: Designing a Data Flow

This data flow consists of a data source object, an SQL object in which the columns to be extracted are defined, and a target object that contains the data target you are loading. Double-click these objects to view the columns of each object. In the SQL column selection object, you can add filtering and sorting criteria, arrange columns, insert columns, and create expressions to generate columns of new data. In the target object, you can perform transformations that apply calculations to columns (for example, you can multiply two columns or use functions to perform specialized calculations), edit synonyms, view sample data, and specify mapping logic.



The navigation pane at the left lists available servers and available synonyms for source and target objects.

A data flow typically becomes an object in a process flow. To open a data flow from a process flow, right-click the data flow and select *Open*. For details about the creation and relationships between data and process flows, see *Designing a Data Flow* on page 171 and *Designing a Process Flow* on page 369.

Revised Functionality

Starting with Release 7.7 Data Migrator has some functionality changes from previous releases. The following topics describe these changes.

Testing the TIBCO WebFOCUS Reporting Server

Prior to release 7.7.07, right-clicking a server name from the DMC opened a shortcut menu that contained a Test Server option. Selecting this option opened a dialog box that contained options for SQL, Focus, and Stored Procedure. The user could select one of these options and then enter a select statement, FOCUS request, or stored procedure to test the server. This option has been removed.

These operations are now done by performing the following steps:

1. Right-click an application directory, such as *foccache*, select *New* and then click *Procedure*.

2. In the procedure area, enter a TABLE request. Alternately, you can type *SQL*, followed by a SELECT statement or EXEC followed by a procedure name.
3. On the Text Editor tab, in the Run group, click *Run*.

When you are done with your testing, you can discard the procedure by performing the following steps:

1. Go to the tab that shows the procedure name, for example, foccache/flow1.fex, and click *X* to close the window.

   The Save Changes dialog box opens.

2. Click *No*.

Quick Copy

In previous server releases, Quick Copy provided the capability to partition the data for faster loading and to select the columns to copy. While these features are no longer available from the DMC, they are available from the *Custom Copy* option on the Web Console.

Keeping the TIBCO WebFOCUS Reporting Server Connection Alive

The Data Management Console normally connects to the server for each request, for example, when running a flow or stored procedure. When the task completes, the connection to the server is severed. This mode of operation is similar to a web browser and helps to keep server resources available to all users.

In previous releases, the Keep Server Connection Alive option kept a constant connection to the server for the duration of the DMC session. This tied up the server resources regardless of whether they were being used or not.

However, this functionality allowed for tasks to be completed such as setting a value for a variable, issuing a SET or a FILEDEF command, or establishing a DEFINE FUNCTION.

These tasks can now be accomplished with the Edit my profile option. Add any needed commands to your profile. Since the profile is run for every connection to the server, they will be in effect for each session.

Note:

- Any changes to variables, or other settings, by running stored procedures will be overridden by the profile when connecting to the server for the next task.
- Global variables can be retained with a setting. For more information, see *How to Make a Global Variable Persistent in a Session* on page 77.

File Listener

In the previous release, an agent based File Listener configured on the Web Console facilitated real time processing. While this allowed processing files to arrive in a specified directory, only Direct Load flows could be used and they could not be scheduled. The flow also did not complete until the agent was terminated, so the log could not be viewed, nor could an email message be sent after processing.

This facility has been replaced in the current release with a new adapter based file listener capability. All configuration is done in a synonym that describes the source files.

The new File Listener has added capabilities. It supports both data flows (SQL based) as well as Direct Load flows and these flows can also be scheduled. The flow completes when a file or time limit is reached, and an email message can be sent on completion.

COMT and TABT Target Files

In previous releases a target with an Adapter of Formatted File could have Format of COMT. Data Migrator would create a comma-delimited file with double quotes around all character columns and the column names as the first line of the file. The TABT format used a tab character as a delimiter instead of a comma.

In Release 7.7 and higher, support for COMT and TABT files has been superseded with the support for Delimited Files that allow any delimiter to be specified, not just comma and tab. The enclosure character for strings can also be specified as any character. Inclusion of the column names as the first line is now optional.

Any data flows created in earlier releases with a target type of COMT and TABT will still run. However, when they are opened in the Data Management Console they will be converted to have a target of Delimited File, with the Header set to yes, Enclosure set to a double quote, and Delimiter set to a comma or tab.

Since previous releases did not permit mapping to a predefined synonym describing a delimited file, they allowed adding columns to the target file in the Target Transformations panel. In the current release, a synonym that describes a delimited file can be used as a target. Any columns needed in the target file can be included in the target synonym and assigned in the Target Transformations panel.

Custom Reports for the Log and Statistics Tables

The capability to customize the standard log and statistics reports has been discontinued.

Run Deferred (Legacy)

The capability to submit flows directly to the deferred execution scheduler has been discontinued.

Instead, use Submit or Submit with options and select *Do not wait for completion* to run at a specific date and time.

Test Run and Run

The Run option has been replaced by Submit, which submits a flow to run on the sever.

Test Run has been replaced by Run on the browser shortcut menu for flows and on the toolbar for flows.

As with Test Run, when a flow is Run data is loaded and the output appears in the DMC console log (and not in the ETL Log or an email message).

However, Run is not subject to the Maximum number of rows for test reports limit found in the Run Options preferences. To limit the number of rows processed, set the limit from the Run with limits dialog box.

Database Default Values

If a database column has a default value assigned in the database, and no value is assigned in Target Transformations, the database will now assign the default value.

In previous releases, fields with no value assigned in Target Transformations had a NULL value assigned for NULLABLE columns and a space or zero for NOT NULL columns.

Note: If a column is described as NOT NULL and has no default value, Data Migrator now requires a value to be assigned, the same as when using a database insert or update statement.

Unassigned Fields

In releases prior to 7.7.04, if no value was assigned to a field in Target Transformations it was assigned an arbitrary value as shown. This value replaced the default value in the database if applicable.

Column Described As	Value Assigned
NULLABLE (MISSING=ON)	NULL
NOT NULL (MISSING=OFF)	Space or zero

In release 7.7.04 the assignment has been changed to correspond with expected relational database behavior:

Column Described As	Default Value in Database	Value Assigned
NULLABLE (MISSING=ON)	NO	NULL
NULLABLE (MISSING=ON)	Yes	Database default value
NOT NULL (MISSING=OFF)	No	Database error
NOT NULL (MISSING=OFF)	Yes	Database default value

To avoid an error from the relational database, assign a value for all NOT NULL fields.

Getting Help

You can access the Data Migrator online help from within the DMC.

Procedure: How to Access Online Help

To access the online help, click the About button arrow and click *Help Topics*.

You can also go directly to the documentation pertaining to this release by clicking the About button arrow and clicking *PDF User's Guide*.

Chapter 2

Setting Up TIBCO Data Migrator

Before you create data flows and process flows in Data Migrator, setup steps must be completed.

The Data Management Console (DMC) is used to perform the major setup and administration tasks that are required before you can build flows. As a developer, you will probably perform most of these tasks yourself.

Some Data Migrator setup tasks may be performed by a server administrator, if that is a separate function in your organization. A server administrator may use the Web Console to perform these tasks, which may vary somewhat based on how and where data is stored at your site. As an alternative, the Web Console can also be used to perform the setup and administration tasks that are available in the DMC.

You can use this chapter as a checklist to ensure that, regardless of the distribution of setup tasks at your site, all of the preparation steps are completed before your design work begins.

In this chapter:

- Setup At a Glance
- Remote TIBCO WebFOCUS Reporting Server Setup At a Glance
- TIBCO Data Migrator Setup
- Initiating TIBCO WebFOCUS Reporting Server Administration Tasks
- Authorizing TIBCO WebFOCUS Reporting Server Usage and Administration
- Managing Application Directories and Configuring the Application Path
- Configuring Data Adapters
- Creating Metadata
- Testing Sample Data and a Process Flow
- Performing TIBCO WebFOCUS Reporting Server Administration Tasks
- Using Secure Shell FTP (SFTP)
- Upgrading TIBCO Data Migrator From Release 4.3 or 5.1
- Upgrading TIBCO Data Migrator From Release 5.2 and Higher
- Upgrading TIBCO Data Migrator to Release 7.7.07 or 82

Setup At a Glance

The two principal components of Data Migrator are the Reporting Server, which includes the Web Console, and the Data Management Console.

Note: The Data Management Console and Server should be from the same release and gen. If they are not, a message appears informing you that the server and client may be incompatible and that some features may not function. It then asks if you want to continue.

1. Install the Server on the platform that will contain your target data (this is the platform you will be moving data to).

   The Server is typically installed on the platform where your data target (a data warehouse, data mart, or another type of target) will reside. Your first task is determining that location. The remaining tasks on this list assume that your source data resides on that same platform.

   Note: At many sites, source data is on a different platform from the one on which the Server is installed. If this is the case at your site, you need to install, configure, and test one or more remote servers, using the process described in *Remote TIBCO WebFOCUS Reporting Server Setup At a Glance* on page 30.

2. For Windows platforms, the DMC is installed with the Reporting server. For non-Windows platforms, you need to install the DMC separately. For more information, see *How to Install the Data Management Console* on page 32.

   The DMC can be installed on each Windows computer that will be used to build flows or on a shared network drive.

3. Start the Server.
4. Start the DMC by opening For Windows servers, the DMC can be found in Information Builders/WebFOCUS 82 Server.
5. Configure one or more nodes in the DMC configuration file.

   This step identifies the name and port of the Server you will be connecting to and adds the server name(s) to the DMC server list. For more information, see *How to Add a Node to Identify the TIBCO WebFOCUS Reporting Server to the Data Management Console* on page 35.

   Tip: If, during the installation of the DMC, you configured it to access the Server, this step is optional. When installing a Windows server, a default LOOPBACK node is configured. This node should not be deleted.

6. Connect to and test the Server from the DMC. You can test that the server is started and able to perform basic data retrieval (against sample data). For more information, see *How to Test the TIBCO WebFOCUS Reporting Server* on page 37. You can also test the server from the Home page of the Web Console.

7. Configure and test data adapters for the source data you will be using from the DMC Adapters folder. The test verifies that the configuration was successful and you can access the data adapter.

   Note: If all of your source data is stored on remote servers, omit this step and Step 8 and proceed to Step 9.

8. Create and test the metadata for the source data from the DMC. The test verifies data retrieval functionality for the specific data sources you are using. You can also create and test the metadata from the Applications page of the Web Console.

   Note: If you are using more than one data source for the *same* adapter, repeat Step 8 as many times as necessary before going to Step 9. If, however, you have data sources in a variety of adapters, repeat Step 7 and Step 8, then proceed to Step 9.

9. Configure and test the data adapter for the data target from the DMC. You can also configure and test adapters from the Adapters folder of the Web Console.

   This step *is necessary* if the data target is of a different database type than the data source(s) or if all of your data is on remote servers. After completing this step, proceed to Step 10.

   Note: If you have more than one data adapter for your data targets, repeat Step 9 as many times as necessary.

   Step 9 *is not necessary* if the data target(s) and data source(s) are of the same database type. Proceed from Step 8 to Step 10.

10. Create and test the metadata for one or more data targets from the DMC. You can also create and test the metadata from the Applications page of the Web Console.

    Note: This step assumes that you are moving data to existing target tables. If you are building new target tables in the DMC, this step is not required.

11. Open and test the sample flow dmdemo, which is provided for this purpose. For each one, run and view the last log to confirm that the Server is configured and properly processing flows. For more information, see *Testing Sample Data and a Process Flow* on page 64.

12. Configure server operations, including security, run-time IDs, and log and statistics tables. See *Performing TIBCO WebFOCUS Reporting Server Administration Tasks* on page 64.

    Note: Authorize any users who will be using Data Migrator from the Data Management Console or Web Console. For details, see *Authorizing TIBCO WebFOCUS Reporting Server Usage and Administration* on page 38.

13. Configure the application path and set up additional application directories. For more information, see *Managing Application Directories and Configuring the Application Path* on page 41. These procedures can also be performed from the Web Console. For more information, see the *TIBCO WebFOCUS® Reporting Server Administration* manual or the Web Console online help.

14. Configure email notification and the scheduler from the Data Management Console. For more information, see *Configuring TIBCO WebFOCUS Reporting Server Operations* on page 65.

For detailed information about the Web Console pages used to set up Data Migrator for local data access, see *Performing TIBCO WebFOCUS Reporting Server Administration Tasks* on page 64 and the *TIBCO WebFOCUS® Reporting Server Administration* manual.

Remote TIBCO WebFOCUS Reporting Server Setup At a Glance

Although the Server is typically installed on the same platform as the target data for DM applications, there is no such restriction for source data. In fact, the Reporting Server frequently resides on one platform while the source data resides on one or more platforms that are remote from the Data Migrator Server itself. In these situations, you must configure the remote servers (also called subservers) to make the data accessible to the Reporting Server for processing. For an illustration of this setup, see the second diagram under *TIBCO WebFOCUS Reporting Server* on page 15.

First, complete the basic Server configuration steps described in *Setup At a Glance* on page 28. After completing these steps, you are ready to start the setup for one or more remote servers.

To set up the remote server, complete these steps:

- ❑ Configure remote servers for data sources and create metadata.
- ❑ Return to the Reporting Server and configure it to access the remote servers so that you can use the data residing there.

The following steps describe this process:

1. Install (if necessary) the remote server on the platform required by your site. (In many situations, the remote server may have already been installed.)
2. Start the DMC by selecting *Information Builders* and then *82 DM Client* from the Windows Start menu.*Information Builders/WebFOCUS 82 Server* For Windows servers, the DMC can be found in *Information Builders/WebFOCUS 82 Server*.
3. Configure one or more nodes for the remote server in the DMC configuration file.

   This step identifies the name and port of the remote server you will be connecting to and adds the server name(s) to the DMC server list. For more information, see *How to Add a Node to Identify the TIBCO WebFOCUS Reporting Server to the Data Management Console* on page 35.

4. Connect to and test the remote server from the DMC. You can test that the server is started and able to perform basic data retrieval (against sample data). See *How to Test the TIBCO WebFOCUS Reporting Server* on page 37. You can also test the server from the Home page of the Web Console.
5. Configure and test the data adapter(s) for the data source(s) you will be using from the DMC. The test verifies that the configuration was successful and you can access the data adapter.
6. Create and test the metadata for the data source(s) you will be using from the DMC. The test verifies data retrieval functionality for the specific data sources you are using. You can also create and test the metadata from the Managing Metadata page of the remote server Web Console.

   Note: If you are using more than one data source for the *same* adapter, repeat Step 6 as many times as necessary before going to Step 7. If, however, you have data sources in a variety of adapters, repeat Step 5 and Step 6, then proceed to Step 7.
7. Connect the Reporting Server to the remote servers. Once you have installed and configured the remote servers, with their data sources, you must establish a connection from the Reporting Server to the remote servers so that Data Migrator will be able to access the data that resides on those servers. To do this, you return to the Web Console Home page of the Reporting Server and select *Remote Servers* from the navigation pane. To add the connection to the remote server(s), you click *Add* in the navigation pane.
8. Create metadata to *mirror* the metadata on the remote server(s) from the Reporting Server Web Console Remote Servers page. (Note that for remote servers, you create metadata from the Remote Servers page rather than from the Metadata page that you used for the Reporting Server.)

   This step makes the data on the remote server accessible to the Reporting Server.

For detailed information about the Web Console pages used to set up Data Migrator for remote data access, see the *TIBCO WebFOCUS® Reporting Server Administration* manual.

TIBCO Data Migrator Setup

In addition to providing a graphical interface for designing flows, the DMC can also be used to perform the major setup and administrative tasks.

Note: When you install a Windows server on a computer where you are building flows, the DMC is installed automatically. For other server types or for access to remote Windows servers, the DMC must be installed separately.

Reference: TIBCO® WebFOCUS Environments

The DMC allows you to configure the mid-tier environment for a WebFOCUS Server.

Reference: Requirements for Installing the Data Management Console

To install the DMC, you need the following:

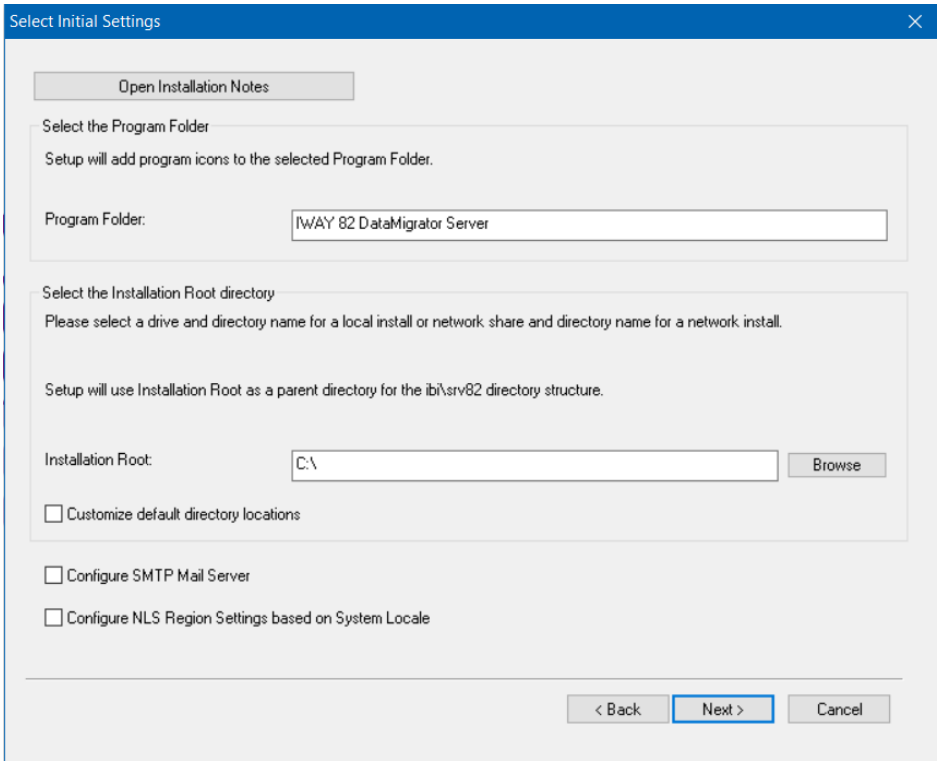
- ❑ Pentium 550 mHz processor or higher.
- ❑ 400 MB of hard disk space.
- ❑ 512 MB of RAM.
- ❑ SVGA or higher resolution graphics card set to at least 800 x 600 pixels and high color (16 bit). (The preferred setting is 1024 x 768, with a color depth of 16 bits.)
- ❑ A mouse or compatible pointing device supported by Windows.
- ❑ A network (TCP/IP) connecting to the Server.
- ❑ Any Windows operating system currently supported by Microsoft®.

Procedure: How to Install the Data Management Console

1. Close any open programs.
2. Unzip the iserver.zip file to a directory on your hard drive. In Windows Explorer, navigate to the directory and run setup.exe.

   The setup program begins and the InstallShield Wizard opens.
3. In the Choose Setup Language window, select a language and click *Next*.

4. The InstallShield Wizard displays the Select the Program Folder and Standard Location Prompt dialog box, as shown in the following image.

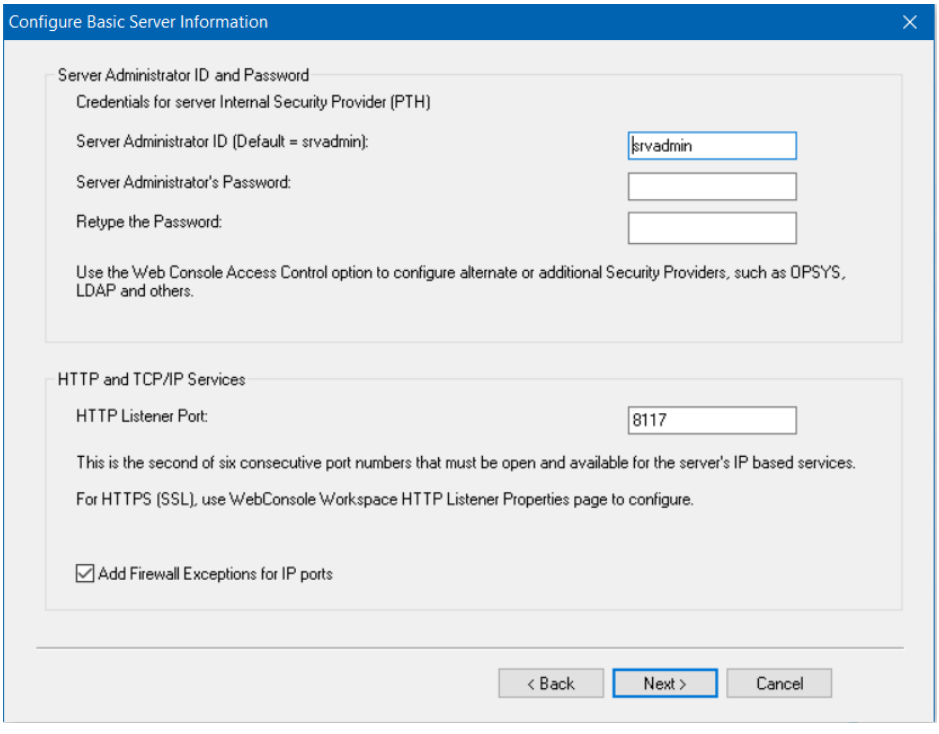


5. The default Program Folder name, as it will appear in the Windows Start menu, is Information Builders 82 Server. If you wish to use a different folder name, type it here. (You can also choose a folder from the Existing Folders list.)

6. Accept the default root and product configuration directories, or browse to directories in which you prefer to install your software. Click *Next*.

   The Configure Basic Server Information dialog box opens.

7. When you install the software, you can set up communication to a server. When you start the software, the server you identify here appears in the Server list in the navigation pane. The following image shows the Configure Basic Server Information dialog box.



- If you enter a user ID and password in the Server Administrator ID and Password boxes, the client saves the entries, setting security to *Explicit* by default. You will not be prompted for them again from the DMC. This step is optional.
- In the HTTP Listener box, enter the port number of the Reporting Server. If you used the defaults when you installed your Server, the HTTP Listener port number is 8121.

  Note: The server will use a group of six ports, starting one less than the port entered here.
- In the Host Name box, type the computer name or IP address where the Reporting Server is running.

8. Click *Next*.
9. Click *Continue*.

   Files are copied to the designated locations. Percent of progress is noted during this process.

Informational messages indicate that setup is configuring the product based on your specifications and that the program folder and icons are being created.

10. Optionally, you can start the Server and Web Console, or start the Data Management Console as soon as the installation is complete.
11. Click *Finish*.

Procedure: How to Launch the Data Management Console and Connect to a TIBCO WebFOCUS Reporting Server

Note: For non-Windows server installations, the DMC can be found in information Builders \WebFOCUS 82 DM Client\Data Management Console. For Windows server installations, the DMC can be found in *Information Builders**WebFOCUS 82 Server*\Data Management Console.

1. On a Windows system, click the *Start* menu, and click *Programs*. If using Windows 10, click *All Apps*.
2. Click *Information Builders*, then WebFOCUS 82 DM Client\Data Management Console or WebFOCUS 82 Server\Data Management Console.

   The DMC opens with the available servers listed in the navigation pane on the left.
3. Right-click the server you wish to use and click *Connect*.

   If you have not entered connection information, the DMC login dialog box opens. Enter the credentials in the fields provided.
4. If your server has security enabled, type your user ID and password in the User Name and Password fields. If you are connecting with a domain user ID, select the domain from the drop-down menu.

   Windows requires that your user name be no longer than 32 characters.

   Note: If you select *Save credentials*, you do not need to enter these credentials each time you connect, and the dialog box does not appear. However, you must reenter this information when you change your password for the server. To do so, right-click the server name in the navigation tree and click *Properties* to reopen the Remote Server Configuration dialog box.
5. Click *Connect*.

   For data retrieval and processing to occur, you must perform this connection step each time you start the DMC.

Procedure: How to Add a Node to Identify the TIBCO WebFOCUS Reporting Server to the Data Management Console

You can set up the DMC to connect to any number of servers in your enterprise.

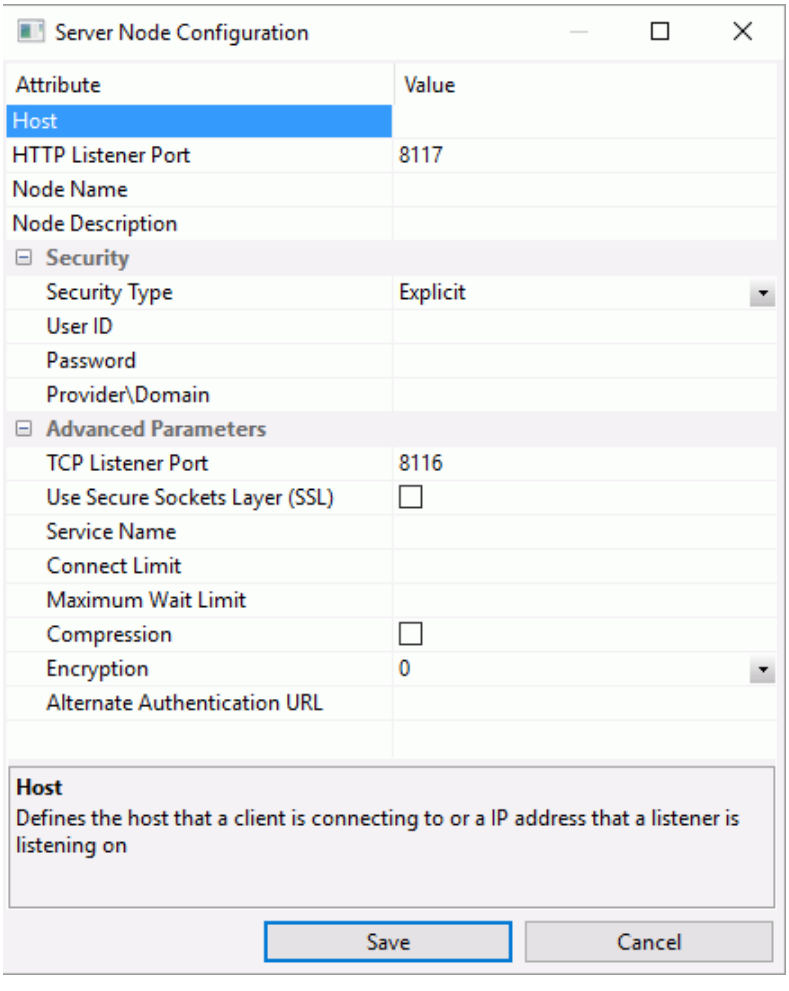
When the DMC opens for Windows servers, the default LOOPBACK node is listed under Servers in the navigation pane. It does not appear if you open the DMC from a non-Windows installation.

Warning: Do not delete the LOOPBACK node, since it is required for internal processing.

To add another server node in this location:

1. Right-click *Servers* and click *Add Server Node*.

   The Server Node Configuration dialog box opens, as shown in the following image.



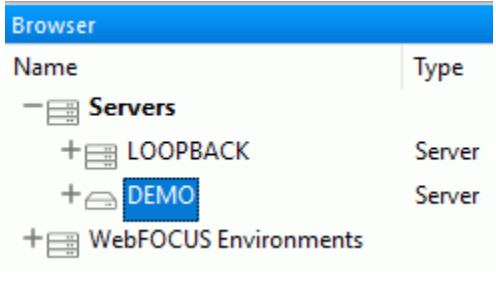
Note: The DMC connects to server nodes using an HTTP listener port and the properties specified in the Host section. Applications that connect using the EDA API, like Resource Analyzer/Governor, use a TCP listener port and the properties specified in the Advanced Parameters section.

2. In the Host box, type the computer name or IP address where the Server is running. For a server running on the same PC as the client, you can type localhost.
3. In the HTTP Listener Port box, enter the port number of the Server, which is 8121 if you used the installation defaults. This port listens to all incoming requests from the DMC and the Web Console.
4. In the Node Name box, type a name for the Server node you wish to add.
5. Optionally, type a description of the node in the Node Description box.
6. Optionally, use a Secured Sockets Layer (SSL) by selecting that check box in the Advanced Parameters section.
7. Security is set to *Explicit* by default. When this setting is in effect, the User ID and password you used to log on to Windows are not passed to the server. If security is required, you must enter it either at connection time or in the Security credentials fields to the right.

   Alternatively, for servers running on Windows, you can choose *IWA* (Integrated Windows Authentication), which passes the Windows User ID and password to the server when a connection is made.

   For information about configuration parameters, see *Viewing Properties* on page 152.
8. Click *OK*.

   The node is added to the Servers tree, as shown in the following image.



You can test the connection between the DMC and the Server you have added.

Procedure: How to Test the TIBCO WebFOCUS Reporting Server

Before you proceed, it is recommended to test the connection between the DMC and the Server.

To ensure that the server is running and that you are able to retrieve sample data:

1. In the DMC, double-click the server name. If you can connect to the server, multiple folders, including Application Directories, Adapters, and others, open in the navigation pane.
2. Expand the *Application Directories* folder.

   If you can see the list of application directories, you are connected to the server.

Initiating TIBCO WebFOCUS Reporting Server Administration Tasks

Although all required server administration and monitoring tasks can be performed by a designated administrator directly from the Server Web Console, an authorized developer can conveniently initiate the more common administration tasks from the DMC using a server administration ID.

To perform any of these tasks, your server must be running so that you can connect to it from the DMC.

Procedure: How to Launch the Web Console

You can access the Web Console directly from the DMC to perform additional server administration tasks. This procedure assumes that the server has already been started.

From the navigation pane, right-click a server name, and click *Access Web Console*.

The Web Console opens in your browser.

Authorizing TIBCO WebFOCUS Reporting Server Usage and Administration

Server administration functions require a connection with a server administration user ID, such as the user ID that was used to install and configure the server. This user can add additional server administration user IDs.

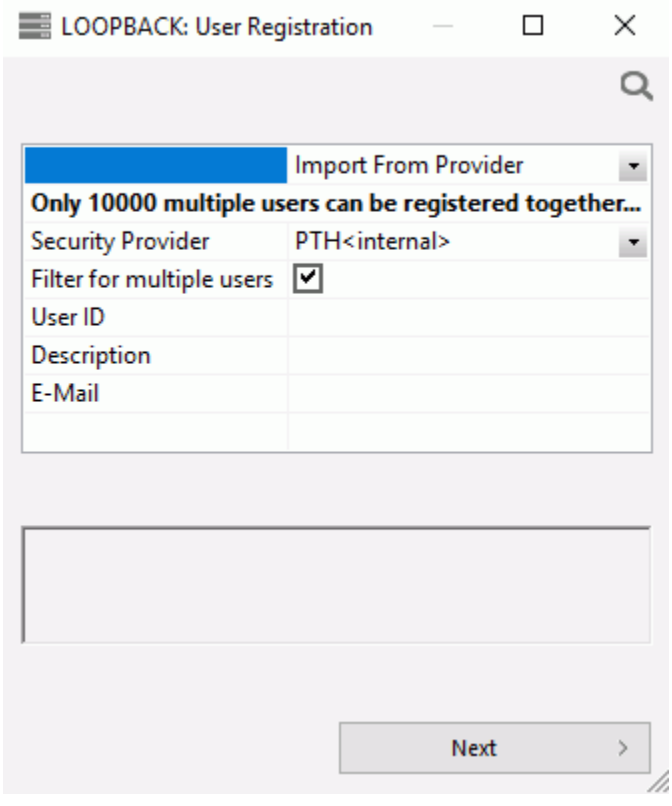
Note: To use Data Migrator from the Data Management Console or Web Console, a user must be at least an Application Administrator.

Procedure: How to Add Administration IDs

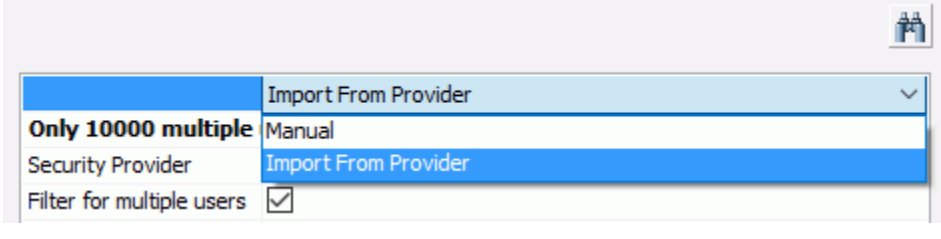
1. In the DMC, expand the *Access Control* folder.
2. Right-click the *Roles* folder and click *Register PTH <internal> User*.

   Note: The Access Control folder options will vary depending on the Security Mode setting. If the Security Mode is set to OFF, no other options will be available.

The User Registration dialog box opens, as shown in the following image.



3. From the drop-down menu, select either *Manual* or *Import from provider*, as shown in the following image.



If you select *Import From Provider*, the security provider will provide the list of users available for registration.

4. If you select *Manual*, complete the following steps:

   a. Fill in the User, Description, and E-Mail boxes.

b. Select a group from the *Inherit Privileges from* drop-down menu:

Server Administrator

Permits all functions.

Server Operator

Permits monitoring of server processes.

Application Administrator

Permits management of flows. This administration level displays flows in the Web Console. It is the minimum level required for DM developers who want to use the Web Console to view their flows and/or to use the DMC to create or edit flows.

Basic User

Provides no administrative or monitoring capabilities.

None

Does not provide privileges.

c. Type values in the Password and Confirm Password boxes.

d. Click *Add and Register*.

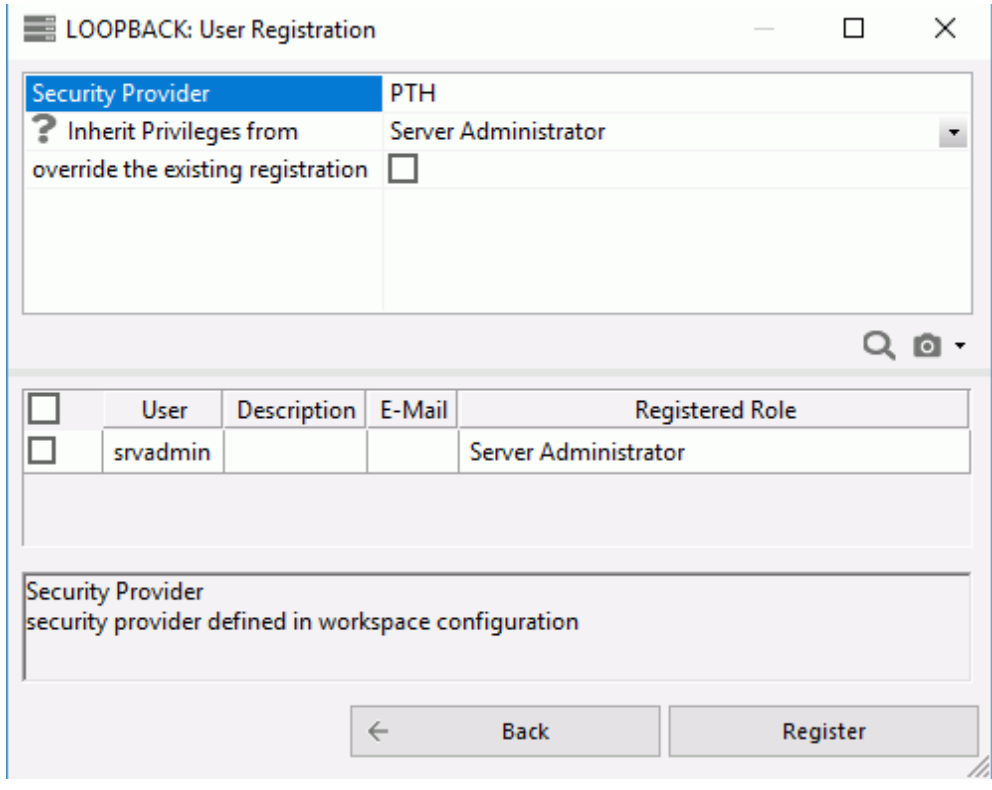
e. Click *OK* in the Confirm window to save your changes and register the new user.

5. If you select *Import from provider*, complete the following steps:

a. Clear the *Filter for multiple users* check box.

b. Click *Next*.

The User Registration dialog box opens, as shown in the following image.



c. Select the checkbox next to a user and double-click the Description and E-Mail boxes to type in values.

d. Click *Register*.

6. In the Activate Providers dialog box, select the status for each security level and click *Save Provider's Status*.

Managing Application Directories and Configuring the Application Path

Procedures and the synonyms they use reside in application directories. The Application Path is a list of viewable application directories. It determines their order of appearance in the navigation pane.

The Application Path is the default search criteria for the server. The server searches each of the directories listed in the Application Path sequentially to find the requested procedure or synonym.

By default, the server profile (EDASPROF.PRF) is run for all users when they connect to the Server to provide access to all application directories in the server search path.

However, you can control a user's access to application directories by creating individual user profiles. Each user can then:

- Access only the application directories specified in the Application Path specified for that profile.
- Use synonyms and procedures in the specified Application Path.

Grouping procedures and synonyms into application directories provides better control. For your convenience, three directories are included in the server search path during installation:

- *baseapp* is included in the search path for every user profile the DM administrator creates. This directory cannot be removed.
- *ibisamp* is the default location for sample files you can use to create procedures, as well as sample procedures.
- *foccache* is a special directory that allows you to create and reuse files for the duration of your Data Management Console session, or until the maximum retention time is reached (set by the server administrator).

You can, of course, set up additional application directories in the server search path.

The following operations are available from the Application Directories folder by right-clicking in the DMC or Web Console.

- Creating sample procedures and data.
- Setting up application directory mappings for different user profiles.
- Configuring the application path by adding, removing, or reordering application directories.
- Filtering the procedures or synonyms that appear in the application directories.
- Creating new procedures and synonyms.

You can also use the Configure Application Path window to set the viewable directories for Data Migrator, as well as add new directories and mappings. To use DM procedures that reside in a directory other than ibi\apps\baseapp, the directory must be added to the list of viewable directories.

You can also manage application directories and configure the Application Path from the Applications page of the Web Console. For more information, see the *TIBCO WebFOCUS® Reporting Server Administration* manual or the Web Console online help.

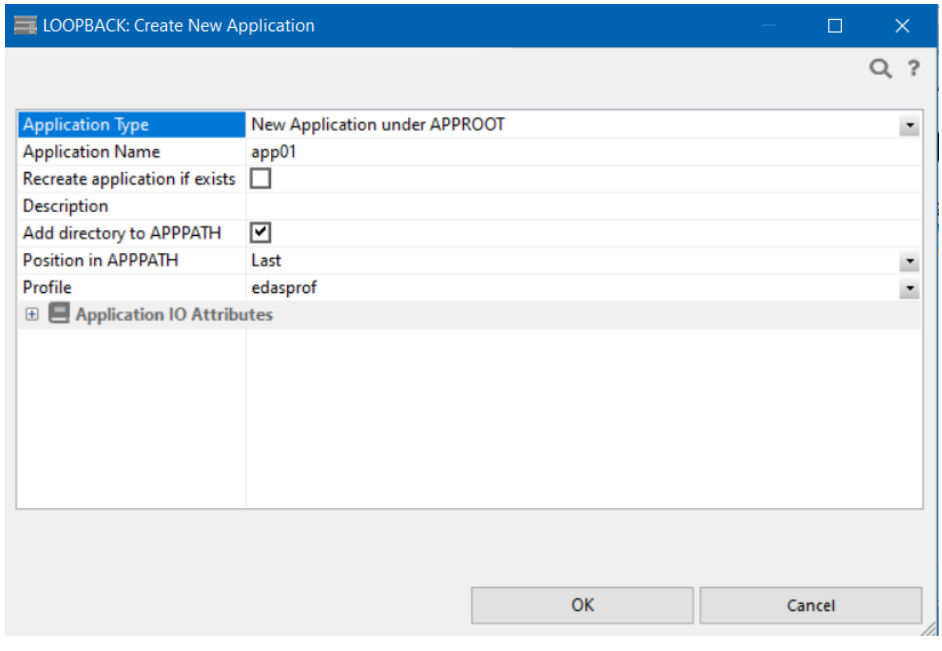
Note: The instructions in the next several sections are for changing the server profile (edasprof.prf), which applies to all users. However, each user can also have their own profile which is used If security is enabled on the server. An Application Administrator can, by default, change their own application path. A Server Administrator can also change the application path of any user.

To change the application path of a user, expand the Access Control and then Roles folder. Expand the user's role folder and right-click on user's ID and click *Application Path*. On the Application Path Configuration panel, click *Change Profile Preference* and select a value other than *Inherit from previously executed profiles*.

Procedure: How to Create an Application Directory

1. In the navigation pane, right-click *Application Directories*, point to *New*, and then click *Application Directory*.

   The Create New Application dialog box opens, as shown in the following image.



2. Enter the name of the new application directory, and, optionally, a description. Leave the *Add directory to APPPATH* check box selected.

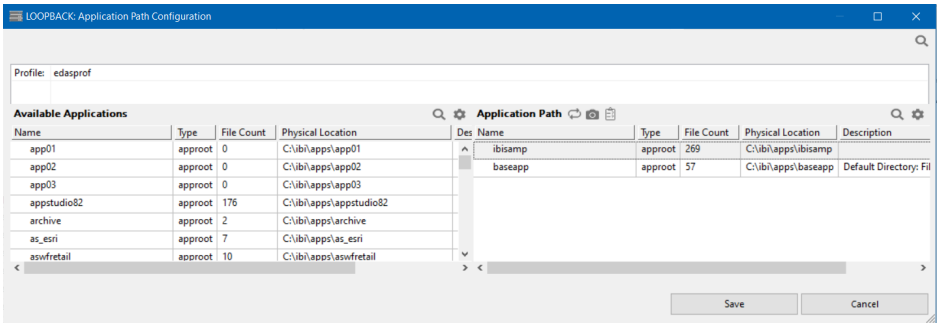
   Caution: The directory name may not contain spaces, special characters, or capital letters. Capital letters are converted to lowercase. If an application directory is created outside of the DMC or Web Console, the same rules apply.

3. Click *OK*.

Procedure: How to Remove an Application Directory From the Application Path

1. Right-click *Application Directories* in the navigation pane and click *Manage*, then *Application Path*.

   The Application Path Configuration dialog box opens, as shown in the following image.



2. Find the application directory you wish to delete. Right-click the path and click *Remove from path*.
3. Click *Save*.

Procedure: How to Delete an Application Directory

Note: You must be connected to a server from a Server Administration ID.

1. If the application directory is still located in the application path, remove it. For more information, see *How to Configure the Application Path* on page 45.
2. Right-click *Application Directories* and click *Application Preferences*.

   ❏ Select the *Show Applications not in PATH* check box.

   ❏ Skip this step if *Hide Application Directories not in PATH* is the available option.

   ❏ If neither option appears, make sure you are connected to the server from a Server Administration ID.
3. Click *Update*.
4. Expand the *Inactive Directories* folder in the navigation pane.
5. Right-click the directory you want to delete and click *Delete*.
6. In the Confirm dialog box, click *OK*.

Procedure: How to Add Application Directory Mappings

You can create a new mapping to a directory anywhere in the file system and include it in the server path. Application directory mappings can be added and deleted on a per profile basis.

1. Right-click *Application Directories* in the navigation pane, point to *New*, and then click *Application Directory*.

   The Create New Application dialog box opens.

2. From the Application Type drop-down menu, select *Application Mapping to Disk*.

   Additional options appear in the Create New Application dialog box.

3. In the Application Name box, type an application name of up to 64 characters.
4. In the Physical location box, enter the full path of the physical directory to be mapped, in the format required by your platform. If needed, select *Create application if it does not exist* from the Map to drop-down menu.
5. Optionally, enter a description in the *Description* field.
6. Optionally, select the *Add directory to APPPATH* check box.
7. If you select *Add directory to APPPATH*, you can also select from the *Position in APPPATH* and *Profile* drop-down menus.
8. Click *OK*.

   The new mapping appears in the Application Configuration dialog box.

Procedure: How to Delete Application Directory Mappings

1. In the navigation pane, right-click *Application Directories* and click *Application Path*.

   The Application Path Configuration dialog box opens.

2. Find the mapping you wish to delete. Right-click it and click *Remove from Path*.
3. Click *Save*.
4. Click *OK* to confirm the deletion.

Procedure: How to Configure the Application Path

You must add an application directory or mapping to the Application Path for it to be accessible.

1. In the navigation pane, right-click *Application Directories* and click *Application Path*.

   The Application Path Configuration dialog box opens.

2. Click *Switch Profile* to select the profile of the user whose Application Path you wish to set.

3. To add a directory or mapping to the application path, right-click it in the Available Applications section and click *Add to Path*.
4. Optionally, right-click any directory or mapping you want to remove from the application path and click *Remove from Path*. This allows you to remove the directory or mapping from the path without deleting it completely.
5. Set the directory search order by dragging a directory or mapping to its desired position.

   You may want to consider moving the ibisamp directory to the bottom of the list to avoid confusion. If you do not plan to use the sample data, you can remove ibisamp from your Application Path completely.
6. When you have added and arranged all necessary directories and mappings, click *Save*.
7. Click *OK* in the confirmation dialog box.
8. Right-click *Application Directories* in the navigation pane and select *Refresh* to see the directory or mapping in the tree.

   Note: If there is a long list of application directories, you can click the *Find* button on the top right or use Ctrl + F to find a text string in the name, physical location, or description.

Reference: Application Directories Shortcut Menu Options

If you right-click *Application Directories* in the DMC navigation pane, the following shortcut menu options are available:

- Refresh.
- Explorer.
- New.
- Get Data.
- Sort by.
- Manage.

Refresh

Refreshes the application directories tree.

Explorer

Opens a Windows Explorer style window on the right side of the Data Management Console that shows detailed information about the currently selected server object.

New

Allows you to add tutorials, flows, procedures, synonyms, and files on the server. For more information on creating same procedures and data using the Tutorials option, see *How to Create Sample Procedures and Data* on page 59.

Get Data

Allows you to upload files or configure adapters.

Sort by

Allows you to sort the Application Tree by name, size, type, or date modified. The default sort is by type.

Manage

Has the following options:

Settings

Opens an Application Settings dialog box that allows you to set settings for the foccache and home application root directories.

External Repository

Application Path

Allows you to add and delete application directories and mappings from the application path.

Schedule and E-mail

Has the following options:

- **Scheduler Agents.** This report runs for the user whose ID value is stored in the sched_scan_id of a Scheduler. The Scheduler must be active to run the Scheduler Agents report. For more information on this report, see *Scheduler Agents Report Columns* on page 458.
- **Scheduled Events.** Opens a dialog box that shows all jobs scheduled to run during a specified time range.

  The Scheduler must be active to run the Scheduler Events report.

  This report shows all flows and procedures scheduled to run during the time period you selected. The Run Date and Run Time columns show their scheduled run times.

  The User ID column shows the user ID used when running the flow or procedure:

sched_run_id

Shows what user ID the flow is run under.

user

The user ID that saved the flow or procedure.

server_admin_id

The first server administrator ID.

- **Force Scheduler Scan.** Forces the scheduler to rescan the directories immediately, in order to find and run any new or changed flows or procedures according to their schedules.

Preferences

Allows you to set preferences for how objects in an application directory appear. You can select whether to show subdirectories last, show applications not in the path, and show object statistics.

Configuring Data Adapters

Data adapters must be configured to allow the Server to access data on different types of databases. Once a data adapter is configured, you can create metadata for the tables or views in the form of synonyms. Synonyms are used to both access the data as sources, and to write the data to targets.

To access data in the DM process, the data adapters you need must be defined to the server. You need to add and configure a data adapter for every database type you want to access.

You can also configure and test data adapters from the Configuring Data Adapters page of the Web Console. For more information, see the *TIBCO WebFOCUS® Reporting Server Administration* manual or the Web Console online help.

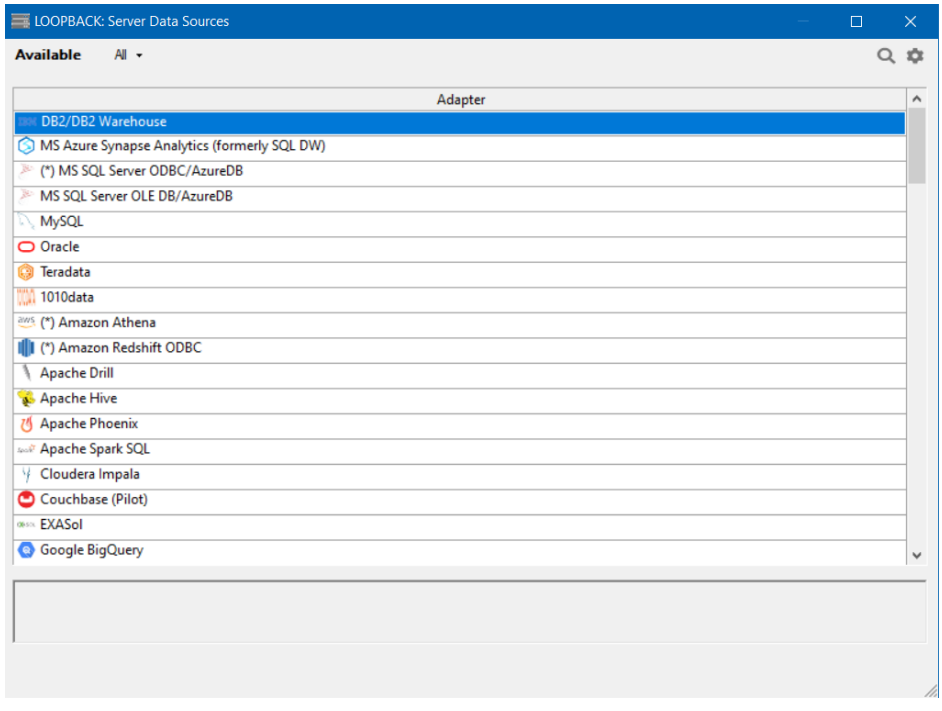
For complete documentation on configuring data adapters and for a detailed explanation of the associated connection parameters, refer to the *TIBCO WebFOCUS® Adapter Administration* manual.

Procedure: How to Add and Configure a Data Adapter

1. Right-click the *Adapters* folder and click *New Datasource*.

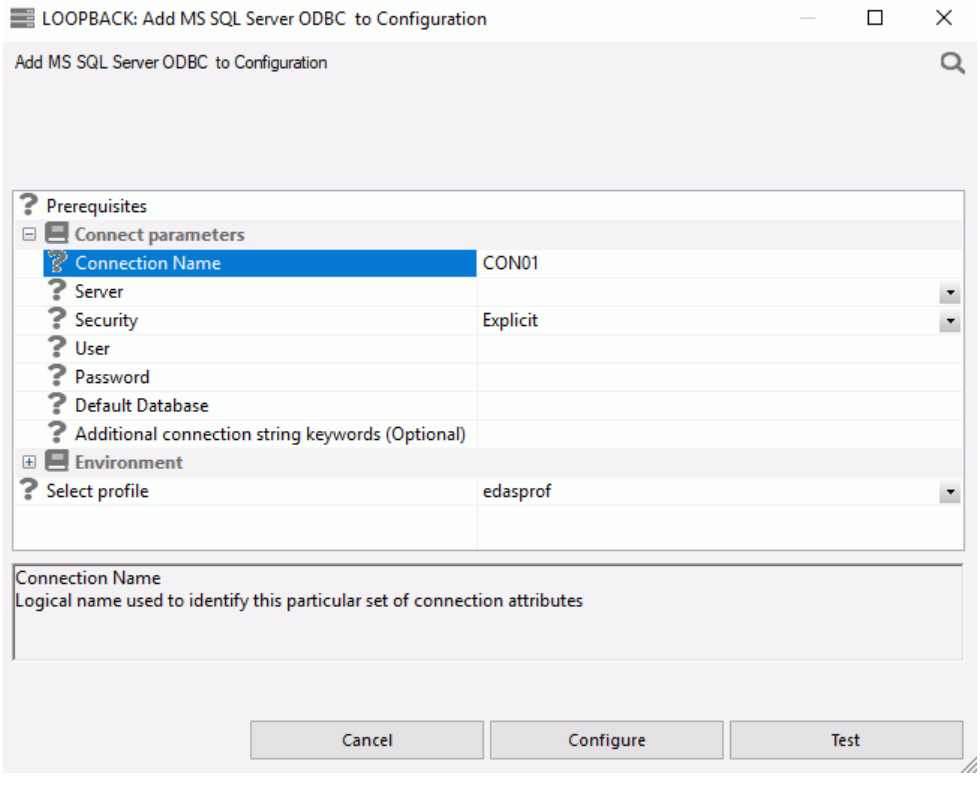
   You can also expand the *Adapters* folder, right-click *Server Data Sources*, and click *New Datasource*

   A dialog box opens listing all of the available adapters, as shown in the following image.



2. Right-click the adapter and click *Configure* or double-click the adapter you want to configure, for example, *MS SQL Server® ODBC*.

The Add to Configuration dialog box opens, as shown in the following image.



3. Configuring the adapter consists of specifying connection and authentication information for each of the connections you want to establish. Enter the appropriate information and click *Configure*.

   A confirmation message appears.

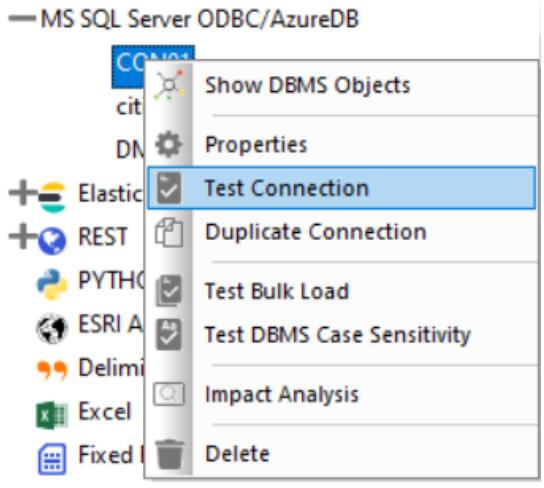
4. Click *Close*.

For complete documentation on configuring data adapters and for a detailed explanation of the associated connection parameters, refer to the *TIBCO WebFOCUS® Adapter Administration* manual.

Procedure: How to Test a Data Adapter

1. Expand the Adapters folder in the navigation pane if it is not already expanded.
2. Expand the folder for the data adapter and right-click the configured adapter connection.

3. Click *Test Connection*, as shown in the following image.

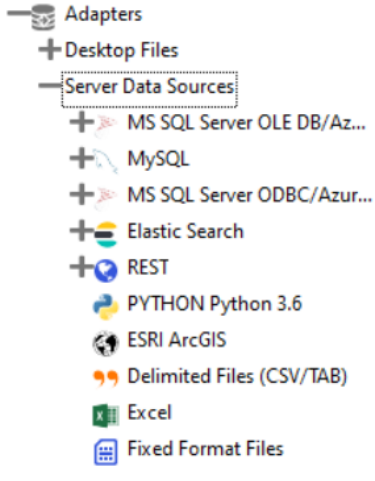


The Test Results open.

Note: The output will vary depending on the adapter.

Reference: **Navigation Pane: Adapters Folder**

The Adapters folder shown in the following image allows you to add and configure adapters from the DMC.



The Adapters folder has the following subfolders:

Desktop files

Contains a list of file types that can be uploaded to the DMC.

Server Data Sources

Contains the adapters that have been configured for this server, as well as the Remote Servers folder.

If you right-click the *Adapters* folder, the following options are available:

New Datasource

Opens a list of adapters that are available to be configured.

Options

Has the following additional options:

Common Settings

Opens the Change Settings for Common Adapter dialog box, which allows you to change the default adapter settings.

Complete Adapter List

Shows a list of data adapters that are available on all platforms.

SQL APT Status

Displays a report of SQL functions and keywords that show how they are handled for each supported database. The keywords are:

- **pass.** Passes the function to the database as is, without doing a conversion.
- **convert.** Converts the function into database specific implementation and passes it to the database.
- **focus.** Forces SQL to be processed by FOCUS.

SQL Optimization Report

Opens the SQL Optimization Report dialog box, which allows you to specify settings for, and run, the report. The SQL Optimization report shows which functions are optimized to their SQL counterparts by adapter.

Data Types

Opens the Filter Data Types Report dialog box, which allows you to specify settings for, and run, the report. The Data Type report contains mapping options for different data types.

Adapters License Count

Lists the number of allowed and configured adapters.

If you right-click an individual adapter folder, the following options are available:

Add connection

Allows you to configure another connection for that adapter.

Change Settings

Allows you to change optimization settings, controlling session parameters, and miscellaneous settings, depending on the adapter.

Remove

Removes the configured adapter from the server.

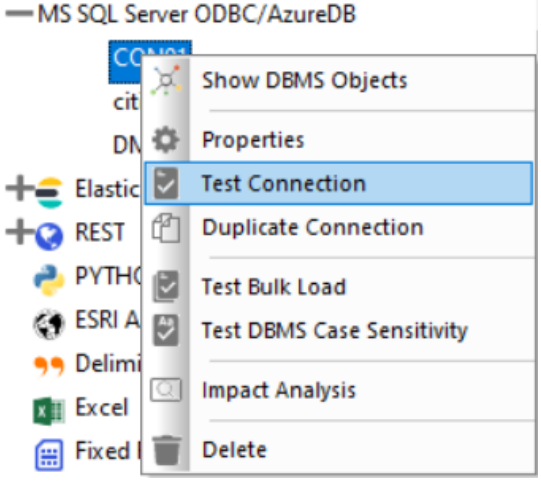
Help

Launches the Web Console Help for the selected adapter.

Prerequisites

Opens a separate page in a browser that lists the prerequisites for the selected adapter.

If you right-click an adapter connection, the options shown in the following image are available:



Show DBMS objects

Allows you to create or update a synonym for that connection.

Properties

Allows you to view and change the connection properties.

Test Connection

Allows you to test the connection.

Duplicate Connection

Allows you to quickly create another connection, using the parameters for the selected connection.

Test Bulk

Performs a bulk load test on the selected connection.

Impact Analysis

Shows synonyms and flows that use the selected connection.

Delete

Allows you to delete the connection.

Creating Metadata

Synonyms constitute the metadata that Data Migrator requires to define unique names (or aliases) for each table or view that is accessible from the server. A synonym must be created for every table that Data Migrator accesses.

Once you have created the metadata (synonym), it is a good idea to test it to make sure that the data is available for DM processing.

You can also create and test metadata from the Managing Metadata page of the Web Console. For more information, see the *TIBCO WebFOCUS® Reporting Server Administration* manual or the Web Console online help.

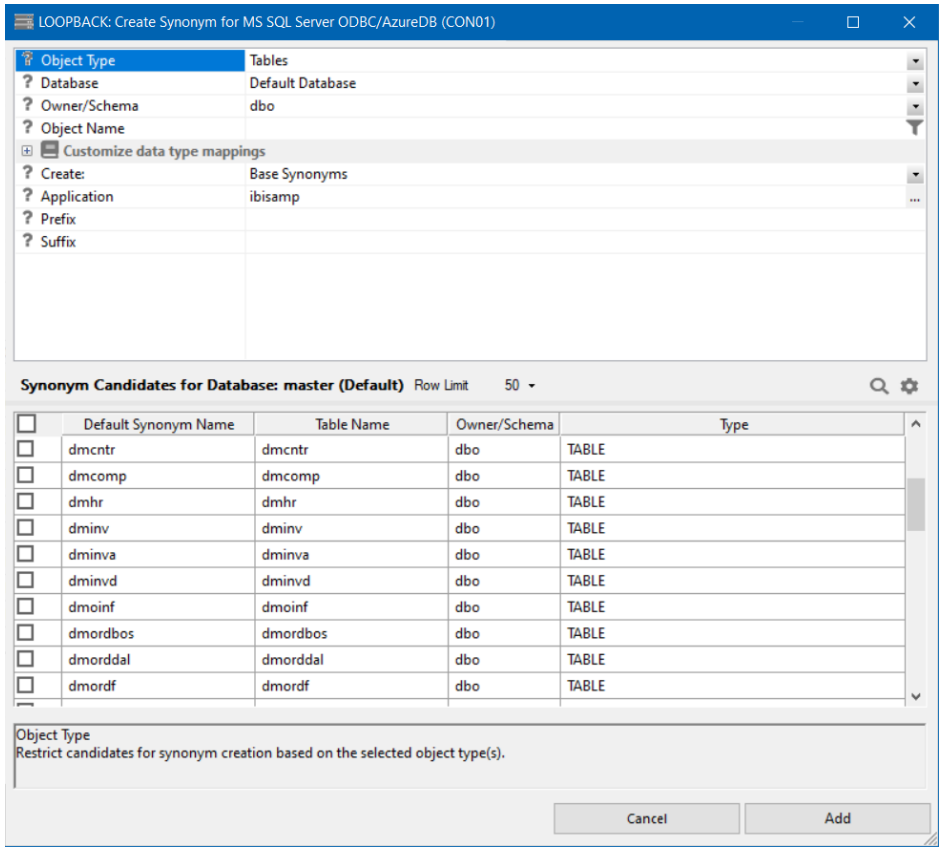
Procedure: How to Create Synonyms for Database Tables

1. In the Data Management Console, right-click an application directory, point to *New*, and then click *Synonym (Create or Update)*.

   The Select dialog box opens, showing the adapters that you have configured.

2. Expand a configured adapter folder, right-click a connection, and click *Show DBMS Objects*.

The Create Synonym dialog box opens, as shown in the following image.



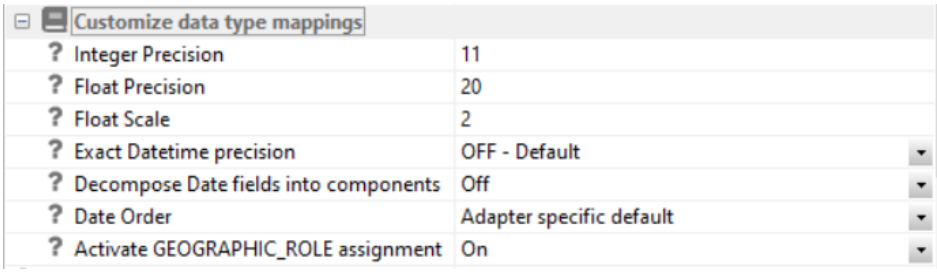
Note: The contents of this dialog box will depend on the adapter selected.

3. To use the database specified when the connection was added, leave *Default Database* as is, or select a different database from the drop-down menu. This option is only available for Informix, Microsoft SQL Server, and Sybase ASE adapters.
4. Optionally, filter the possible candidates by name and owner. Selecting this option adds Owner and Table Name parameters to the dialog box.

   - For the *Owner* parameter, type a string for filtering the owner IDs and insert the wildcard character (%) as needed at the beginning and/or end of the string. For example, enter ABC% to select tables or views whose owner IDs begin with the letters ABC, %ABC to select tables or views whose owner IDs end with the letters ABC, or %ABC% to select tables or views whose owner IDs contain the letters ABC at the beginning, middle, or end.

- For the *Table Name* parameter, type a string for filtering the table or view names and insert the wildcard character (%) as needed at the beginning and/or end of the string. For example, enter ABC% to select all tables or views whose names begin with the letters ABC, %ABC to select tables or views whose names end with the letters ABC, or %ABC% to select tables or views whose names contain the letters ABC at the beginning, middle, or end.

5. If you have tables with identical table names, you can assign a prefix or a suffix to distinguish them. For example, if you have identically named human resources and payroll tables (most likely with different owners), assign the prefix HR to distinguish the synonyms for the human resources tables. Note that the resulting synonym name cannot exceed 64 characters. Alternately, you can specify different application directories for different schemas or owners.
6. the *Customize data type mappings* to access additional settings, as shown in the following image.



7. Optionally, change the scale and precision settings.
8. From the Decompose Date fields into components drop-down menu, select *Off*. This option must be off when using Data Migrator.
9. Complete your table or view selection:

   To select all tables or views in the list, select the check box to the left of the *Default Synonym Name* column heading.

   To select specific tables or views, select the corresponding check boxes.
10. The *Default Synonym Name* column displays the name that will be assigned to the synonym. To assign a different name, replace the displayed value.
11. Click *Add*.

    Synonyms are created and added under the specified application directory.

    The status field of the Create Synonym window shows that the synonyms were created successfully.
12. Click *Close*.

Procedure: How to Create a Synonym for a Relational Stored Procedure

If you create a synonym for a relational stored procedure, you can use that stored procedure as a data source in a flow. When you create the synonym, you will be prompted for values for the parameters of the procedure. For information on creating the flow, see *How to Use a Relational Stored Procedure as a Data Source* on page 185.

We are using Oracle® as an example for this procedure.

1. In the navigation pane, right-click a configured relational Oracle connection and click *Create or Update Synonym*.

   The Select Synonym Candidates (Step 1) dialog box opens.

2. Select *Stored Procedure* from the drop-down menu and the *Filter by owner/schema and object name* check box. Enter an Owner/schema or Object name or both and click *Next*.

   The Select Synonym Candidates (Step 2) dialog box opens.

3. Select the radio button for a stored procedure and click *Next*.

   The Select Synonym Candidates (Step 3) dialog box opens.

4. Select check boxes for the parameters and supply a value for testing. Click .

   Note: Values entered are used only when creating the synonym and are not saved.

   A confirmation message appears.

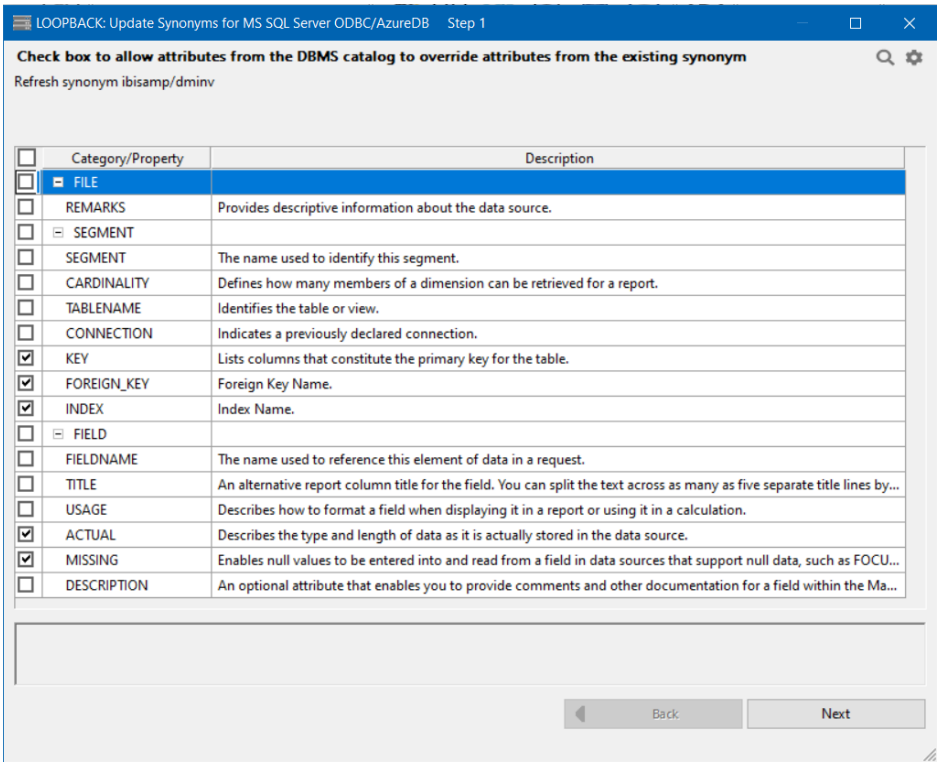
5. Click *Close*.

   The synonym now appears in the navigation pane listed under the application directory it was created in.

Procedure: How to Update a Synonym

1. Right-click a synonym in the Navigation pane, point to *Metadata Management*, and select *Refresh*.

The Update Synonyms dialog box opens, as shown in the following image.



2. Select the attributes that you want to override those from the existing synonym and click *Next*.

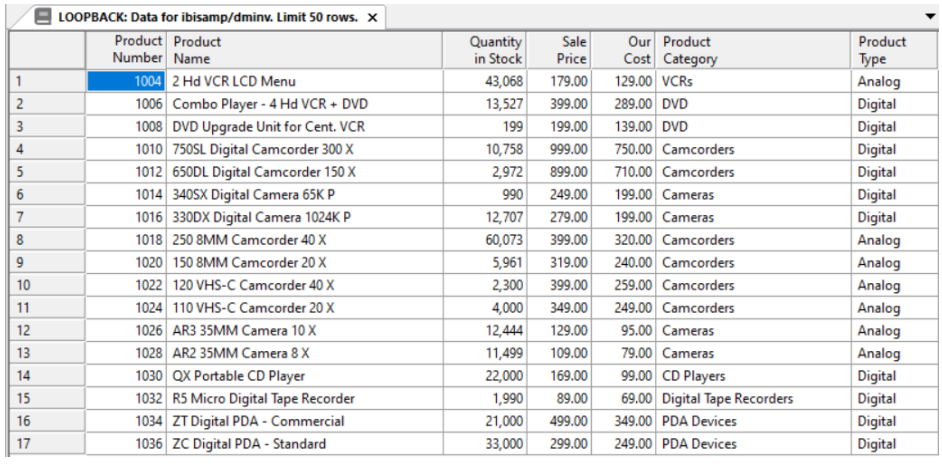
   The selected synonym will now be updated

Procedure: How to Test Synonyms for Tables or Files

1. In the Data Management Console, right-click a server name and click *Connect*.
2. Expand the *Application Directories* folder, and expand the application directory where you created your synonyms.

   Note: If the synonyms are not listed, on the Home tab, from the Filter group, click *Synonyms*.

3. Right-click the synonym you wish to test and click *Sample Data*. Sample data appears in the workspace, and looks similar to the following image.



LOOPBACK: Data for ibisamp/dminv. Limit 50 rows.

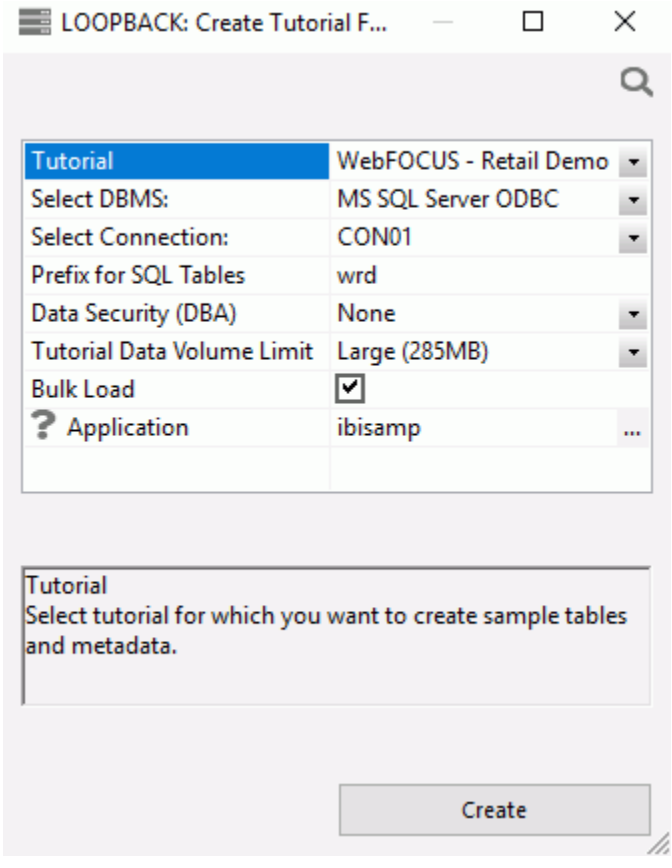
	Product Number	Product Name	Quantity in Stock	Sale Price	Our Cost	Product Category	Product Type
1	1004	2 Hd VCR LCD Menu	43,068	179.00	129.00	VCRs	Analog
2	1006	Combo Player - 4 Hd VCR + DVD	13,527	399.00	289.00	DVD	Digital
3	1008	DVD Upgrade Unit for Cent. VCR	199	199.00	139.00	DVD	Digital
4	1010	750SL Digital Camcorder 300 X	10,758	999.00	750.00	Camcorders	Digital
5	1012	650DL Digital Camcorder 150 X	2,972	899.00	710.00	Camcorders	Digital
6	1014	340SX Digital Camera 65K P	990	249.00	199.00	Cameras	Digital
7	1016	330DX Digital Camera 1024K P	12,707	279.00	199.00	Cameras	Digital
8	1018	250 8MM Camcorder 40 X	60,073	399.00	320.00	Camcorders	Analog
9	1020	150 8MM Camcorder 20 X	5,961	319.00	240.00	Camcorders	Analog
10	1022	120 VHS-C Camcorder 40 X	2,300	399.00	259.00	Camcorders	Analog
11	1024	110 VHS-C Camcorder 20 X	4,000	349.00	249.00	Camcorders	Analog
12	1026	AR3 35MM Camera 10 X	12,444	129.00	95.00	Cameras	Analog
13	1028	AR2 35MM Camera 8 X	11,499	109.00	79.00	Cameras	Analog
14	1030	QX Portable CD Player	22,000	169.00	99.00	CD Players	Digital
15	1032	R5 Micro Digital Tape Recorder	1,990	89.00	69.00	Digital Tape Recorders	Digital
16	1034	ZT Digital PDA - Commercial	21,000	499.00	349.00	PDA Devices	Digital
17	1036	ZC Digital PDA - Standard	33,000	299.00	249.00	PDA Devices	Digital

Procedure: How to Create Sample Procedures and Data

Sample procedures and data are needed for *Tutorial: Creating Data and Process Flows* on page 415 and for the exercises in *Advanced Tips and Techniques* on page 541.

1. From the DMC, right-click *Application Directories* in the navigation pane, point to *New*, and click *Tutorials*.

The Create Tutorial Framework dialog box opens, as shown in the following image.



2. From the Tutorial drop-down menu, select one of the following options:

   - ❏ WebFOCUS - Retail Demo
   - ❏ WebFOCUS - State Population Demo
   - ❏ WebFOCUS - Custom SQL Security Provider
   - ❏ Data Migrator - General
   - ❏ Data Migrator - Iterator
   - ❏ Data Migrator - File Listener
   - ❏ Data Migrator - Star Schema

- ❑ Create Legacy Sample Tables and Files

Note: The *WebFOCUS* and Legacy sample tables and files are not used with Data Migrator.

The *Retail Demo* option loads tables with a pre-determined amount of data designed for WebFOCUS reports.

3. From the *Select DBMS* drop-down menu, choose a DBMS type, and click *Create*. DATREC, a binary file format, will be the only choice if you do not have other adapters configured.

   For the Prefix for SQL Tables field, use the default value unless you have a reason to avoid tables starting with the indicated default. If so, type a different value.

   For the Application directory field, use the default, ibisamp directory to make it easier to follow the tutorial described in *Tutorial: Creating Data and Process Flows* on page 415, *Advanced Tips and Techniques* on page 541, and other examples in this manual.

   When the creation process is complete, a confirmation message appears. The message will depend on the type of samples you create, for example, Data Migrator - General, Data Migrator - Iterator, Data Migrator - File Listener, or Data Migrator - Star Schema.

   The message also displays in the Console Log.

   The procedures and synonyms appear in the *ibisamp* application directory, which is created and added to the server search path during installation. For FOCUS, the sample data is also stored in *ibisamp*. For other adapters, it is stored in the database of the configured adapter. The source data for the file listener flow is stored in ibisamp in four files with names starting with dmord and an extension of data.

4. Ensure that the ibisamp application directory is in the server path. It will be in the path unless you delete it.

   - ❑ In the DMC, examine the navigation tree in the server you are using.
   - ❑ On the Applications tab of the Web Console, click the *Application Directories* link in the navigation pane to see the application path. For more information, see *How to Configure the Application Path* on page 45.

Procedure: How to Create Column Names in Lower Case

Note: By default, column names are created in the database with UPPER CASE column names. This is problematic for databases, such as PostgreSQL and those based on it such as Greenplum, where column names are expected to be in lower case.

1. In the DMC, expand the *Adapters* folder.
2. Right-click the folder name of the adapter you wish to configure and click *Change Settings*.

The Change Settings dialog box opens.

3. In the Metadata section, from the ALIAS_CASE drop-down menu, select *LOWER - Enforce lower case*.
4. Click *Save*.

This setting is also used by Data Migrator when creating new tables.

Procedure: How to Create Sample Procedures and Data for Star Schema

Star Schema sample procedures are needed for *Loading a Star Schema* on page 605.

1. From the DMC, right-click *Application Directories* in the navigation pane, point to *New*, and then click *Tutorials*.

   The Create Tutorial Framework dialog box opens. Your display will depend on the specific adapters you have configured.
2. Select *Data Migrator - Star Schema* from the Tutorial drop-down menu.
3. From the Select DBMS drop-down menu, select a relational database that you have configured.

   For the Prefix for SQL Tables field, use the default value unless you have a reason to avoid tables starting with the indicated default. If so, type a different value.

   For the Application directory field, use the default, ibisamp directory to make it easier to follow the tutorial described in *Tutorial: Creating Data and Process Flows* on page 415, *Advanced Tips and Techniques* on page 541, and other examples in this manual.
4. Click *Create*. The following status message should appear:

   Star Schema sample procedures created for *database*

   where:

   database

   indicates the database type you have selected.

The procedures and synonyms appear in the ibisamp application directory, which is created and added to the server search path during installation.

Procedure: How to Create Sample Procedures and Data for File Listener

From the DMC:

1. Right-click *Application Directories* in the navigation pane, select *New*, and then click *Tutorials*.

   The Create Tutorial Framework dialog box opens. Your display will depend on the specific adapters you have configured.

2. Select *Data Migrator - File Listener* from the Tutorial drop-down menu.
3. From the Select DBMS drop-down menu, select a relational database that you have configured.

   For the Prefix for SQL Tables field, use the default value unless you have a reason to avoid tables starting with the indicated default. If so, type a different value.

   For the Application directory field, use the default, ibisamp directory to make it easier to follow the tutorial described in *Tutorial: Creating Data and Process Flows* on page 415, *Advanced Tips and Techniques* on page 541, and other examples in this manual.
4. Click *Create* to create the sample procedures and data for file listener.

Procedure: How to Create Sample Procedures and Data for Iteration

Additional sample procedures and data are needed for *Iterating Through Multiple Tables* on page 612. These instructions should be followed after you have loaded the sample data and procedures for Data Migrator.

1. From the DMC, right-click *Application Directories* in the navigation pane, point to *New*, and then click *Tutorials*.

   The Create Tutorial Framework dialog box opens. Your display will depend on the specific adapters you have configured.
2. Select *Data Migrator - Iterator* from the Tutorial drop-down menu.
3. From the Select DBMS drop-down menu, select a relational database that you have configured.

   For the Prefix for SQL Tables field, use the default value unless you have a reason to avoid tables starting with the indicated default. If so, type a different value.

   For the Application directory field, use the default, ibisamp directory to make it easier to follow the tutorial described in *Tutorial: Creating Data and Process Flows* on page 415, *Advanced Tips and Techniques* on page 541, and other examples in this manual.
4. Click *Create*. The following status message should appear:

   Sample procedures created for *database*

   where

   `database`

   Indicates the database type you have selected.

   The procedures and synonyms appear in the ibisamp application directory, which is created and added to the server search path during installation.

5. Load the created tables and expand the *ibisamp* directory.

   Right-click the *dmpIntpf* flow and click *Submit*.

Testing Sample Data and a Process Flow

Once you have verified that all required DM setup tasks have been successfully completed, you can perform three simple tests to ensure that your flows will run properly. We created some sample data and process flow procedures for you to use in testing your setup.

We recommend that you run these tests before you create and attempt to run your own flows.

Example: Testing the Sample Data Flow DMDEMO

1. Expand the *ibisamp* directory in the navigation pane.
2. Double-click the *dmdemo* data flow. The data flow opens in the workspace at the right.

   This data flow extracts data from the data sources named dmord and dminv and loads an existing table named dmrpts.
3. From the Flow tab, in the Run group, click *Run*.
4. Click *Submit* from the drop-down menu.
5. When the data flow run has been completed, the following message appears in the Server Console Log:

   ```
   (ICM18763) Request ibisamp/dmdemo complete
   ```

6. Right-click the *dmdemo* data flow from the ibisamp directory, point to *Logs*, and then click *Log and Statistics*.

   The Log and Statistics dialog box opens. From the *Report Type* drop-down menu, select *Log* and then click the *View Report* button. The log, which is displayed in the workspace, indicates that the table named dmrpts has been loaded with 178 rows.
7. From the Home tab, in the Windows group, click *Windows* and then click *Close All* to close the log and the data flow window.

Next Steps: If you have received these results, you can feel confident that your client/server setup is running correctly and you can proceed to the design phase.

If you are not familiar with DM processing, we urge you to read *Planning a TIBCO Data Migrator Application* on page 87, before you begin creating flows to ensure that you make optimal decisions about such issues as data structures and security to meet the needs of your site.

Performing TIBCO WebFOCUS Reporting Server Administration Tasks

You can control many key aspects of Server operations directly from the DMC.

The DMC allows you to:

- Configure the application path.
- View and run flows by expanding the approot folder and its application directories.
- Create sample procedures and data.
- Set up log and statistics options.
- Run reports.
- View run-time information.
- Schedule flows.
- Set Scheduler settings.
- Set email notification.

Configuring TIBCO WebFOCUS Reporting Server Operations

You can set up your Server to send an email message that notifies recipients of the successful completion or failure of every flow execution.

In addition, from the DMC, you can enable the distribution of email messages at any point in a process flow.

However, to enable either variation on the email notifications process, you must first configure a SMTP service. (You need to set up your server for email before it will work anywhere in the flow.)

Note: You must be logged on as a Server Administrator with the appropriate privileges to make these configurations.

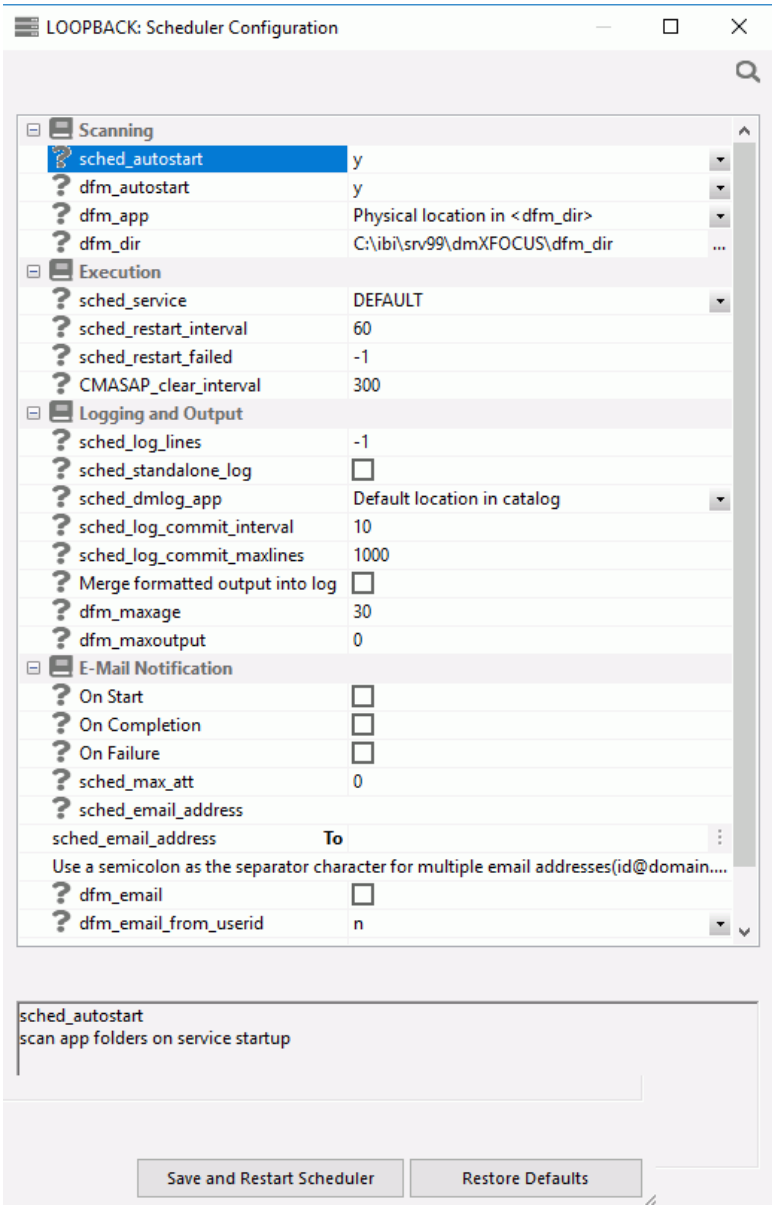
Procedure: How to Edit the Scheduler Configuration

1. In the DMC, expand the *Workspace* folder, followed by the *Special Services and Listeners* folder.
2. Right-click *SCHEDULER* and click *Properties*.

   The Scheduler Configuration dialog box opens.
3. Make your changes, and click *Save and Restart Scheduler*. For more information on the available options, see *Scheduler Configuration Dialog Box* on page 66.

Reference: Scheduler Configuration Dialog Box

The Data Migrator Scheduler runs as a task under the Server. The Scheduler must be running to submit flows and stored procedures and to execute scheduled data flows and flows that run in parallel groups in a process flow. The following image shows the Scheduler Configuration dialog box.



The Scheduler Configuration dialog box contains the following fields/options:

Scanning

sched_autostart

Specifies whether the Scheduler should check for flows automatically. The options are:

y

Automatically starts the Scheduler when the server starts. This is the default value.

n

Does not start the Scheduler when the server starts.

Note: The Scheduler must be running to execute not only scheduled data flows, but also restarted flows and flows run in parallel in a process flow.

dfm_autostart

If this option is enabled, the DFM listener will start with the server.

dfm_app

Specifies the application name of the location of deferred requests and reports.

dfm_dir

Specifies the physical location of the deferred requests and reports, if the dfm_app value is not set.

Execution

sched_service

This is the agent service that the Data Migrator Scheduler will use when running DM flows. If this is not set, the DEFAULT service will be used.

sched_restart_interval

Specifies the time interval in seconds between restart attempts for all restartable flows. The default value is 60.

Note: This value cannot be less than 0.

sched_restart_failed

Specifies the number of times that failed agents will be restarted. If the value is 0, they will not be restarted. If it is -1, they will be restarted indefinitely. The default value is -1.

CMASAP_clear_interval

Specifies how often CMASAP agents are removed from the agents list, in seconds. When set to -1, this is never. The default value is 300.

Logging and Output

sched_log_lines

Maximum number of lines that the Data Migrator Scheduler is to write to the log for each flow. When the maximum number of lines is reached, logging stops. When set to -1, the number of lines is unlimited. When set to 0, there is no logging.

The default value is set to 20.

Note: Some essential lines will still be imposed into the ETL LOG, even when the number of lines (sched_log_lines) is limited to a positive integer. For more information, see *List of Essential ETL LOG Message Codes* on page 677.

sched_standalone_log

Select this option to have a separate log file for every job.

sched_dmlog_app

The name of the application in which the individual log and statistics files are located. This option is only applicable if you select sched_standalone_log.

sched_log_commit_interval

Specifies the maximum time interval in seconds that can elapse before log data is committed to the Data Migrator log. The default value is 10.

sched_log_commit_maxlines

Specifies the maximum number of log lines to collect before committing them to the Data Migrator log. The default value is 1000.

Merge formatted output into log

Select this option to set the output from a report to go to the log. Only text output is displayed to a maximum width of 80 characters. This is the prior behavior.

Note: To view report output, in the browser, right-click on the flow name and click *Logs*. Then, click *Last Output* to view the output from the last run, or select *Log and Statistics*. For Report Range select *All*, and then right-click the job id in the report and click *Show Output*.

dfm_maxage

Specifies the maximum number of days that deferred reports are kept.

dfm_maxoutput

Specifies the maximum size of a deferred report.

E-mail Notification

On Start

Sends an email when the Scheduler starts.

On Completion

Sends an email when the schedule completes.

On Failure

Sends an email if the schedule fails.

sched_email_address

Enter an email address, user ID, or procedure name.

dfm_email

If this option is enabled, the schedule specified for server wide email will be followed.

dfm_email_from_userid

If this option is enabled, the email address associated with the user ID will be retrieved from the security provider.

Note: Before you can use email notification, you must configure an email server. For more information, see *How to Configure the TIBCO WebFOCUS Reporting Server for Email Notification* on page 71.

Procedure: How to Start or Stop the Scheduler

1. From the DMC, connect to the server with an administrator user ID.
2. Expand the *Workspace* folder, and then expand the *Special Services and Listeners* folder.
3. Right-click *SCHEDULER*.

   If the Scheduler is running, select *Stop*.

   If the Scheduler is not running, select *Start*.

Procedure: How to Update Passwords

APPLICATION-level administrators can change their configured passwords from the DMC.

1. In the DMC, right-click the server name, point to *My Console*, and then click *Change Password*.

   The User Information dialog box opens.

2. Enter your current operating system password in the *Current Password* field.
3. Enter the password in the *New Password* and *Confirm Password* fields.
4. Select the *Update operating system password* or *Update password in admin.cfg, server will be restarted* check box.
5. Click *Update*.

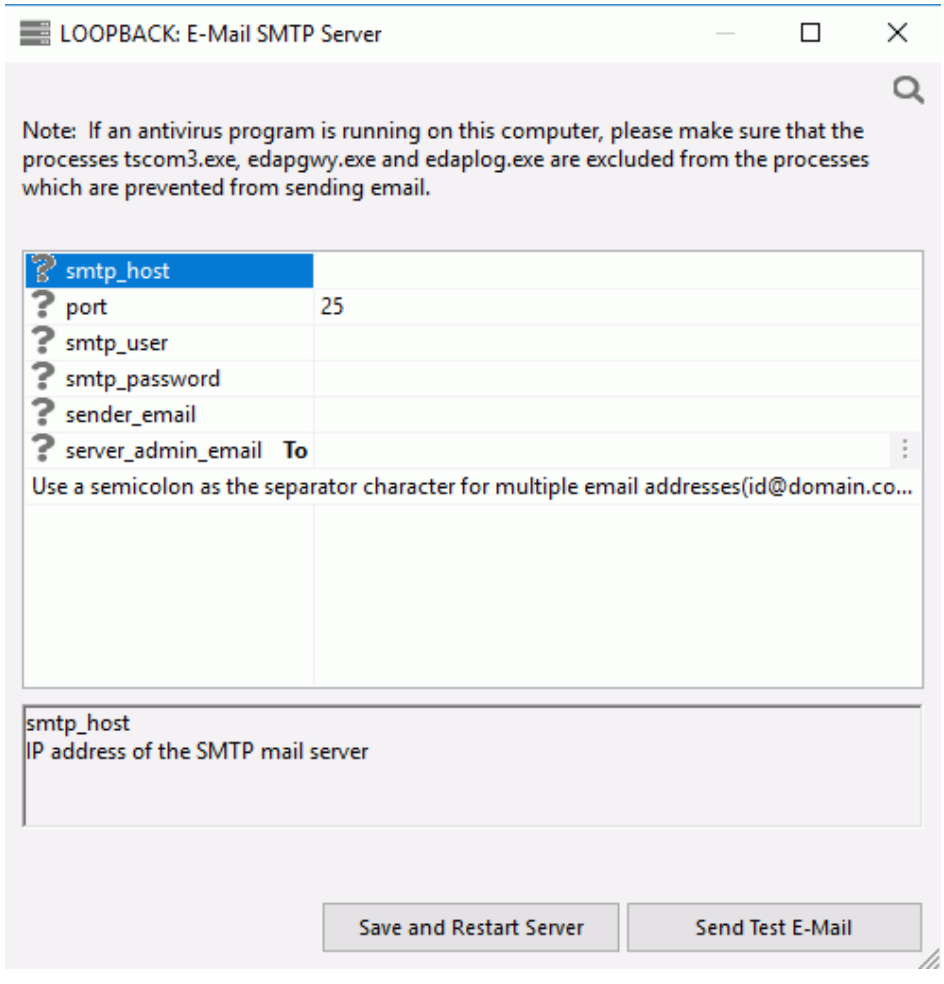
Note: Changes to the password will take effect the next time the server or scheduler is restarted.

Procedure: How to Configure the TIBCO WebFOCUS Reporting Server for Email Notification

To enable email notifications, you must first configure an SMTP service. (You need to set up your server for email before it will work anywhere in the flow.)

1. In the DMC, right-click the *Workspace* folder, point to *Workspace SET*, and click *SMTP Server Settings*.

   The E-mail SMTP Server dialog box opens, as shown in the following image.



   Note: If an antivirus program is running on this computer, please make sure that the tscom3.exe, edapgwy.exe, and edaplog.exe processes are excluded.

2. Enter the following information:

   - **smtp_host.** Defines the IPv4 address of the SMTP email server. The address can be in

standard dot notation (for example, *smtpmail.yourcompany.com*).

- **port.** Specifies that the mail server is bound to the TCP port that is shown.
- **smtp_user.** The user ID to the SMTP server. This value is optional.
- **smtp_password.** The password to the SMTP server. This value is optional.
- **sender_email.** Defines the email address of the sender, which appears in the 'From' field in emails (for example, *yourname@yourcompany.com*).
- **server_admin_email.** Defines the email address of the recipient (for example, *yourname@yourcompany.com*).

3. Click *Save and Restart Server*.
4. After configuring SMTP services, go to the Web Console Scheduler Configuration page and select a sched_email option:

off

Never send email.

completion

Send email when a flow finishes.

failure

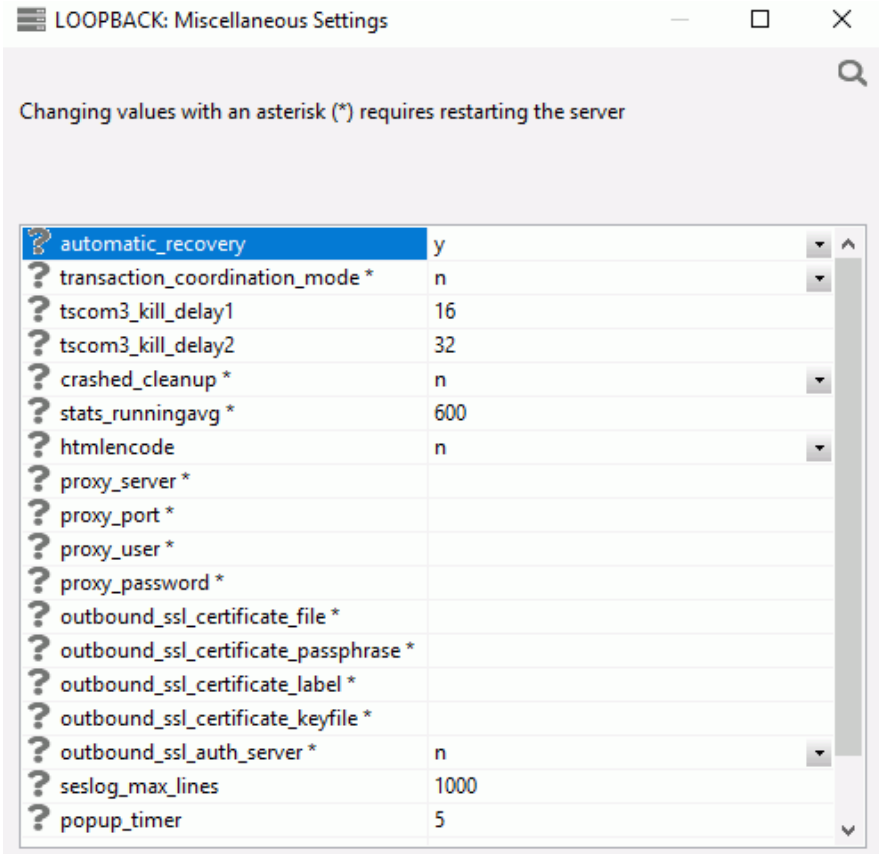
Send email when a flow fails.

Note: This setting applies to every flow.

Procedure: How to Edit the TIBCO WebFOCUS Reporting Server Configuration

1. In the DMC, right-click the *Workspace* folder, point to *Workspace SET*, and then click *Miscellaneous Settings*.

The Miscellaneous Settings dialog box opens, as shown in the following image.



The following Miscellaneous Settings are saved in the edaserv.cfg file:

automatic_recovery

Determines whether the server will automatically recover crashed listeners and/or special services. The default is *n*.

transaction_coordination_mode

Selecting *y*, activates the Transaction Coordinator. A Two-Phase Commit will be used when XA-Compliant databases participate in a transaction. The default is *n*.

tscom3_kill_delay1

Defines the time (in seconds) allowed for an agent to stop before soft kill is sent. The default is 16.

tscom3_kill_delay2

Defines the time (in seconds) allowed for a soft kill to work before a hard kill is sent. The default is 32.

crashed_cleanup

By default, crashed agents are kept in the agents table for diagnostics purposes, and are only removed when they are explicitly killed. When this set to *y*, crashed agents are automatically removed from agents list when the Workspace Manager detects a crash. The default value is *n*.

crashed_max_dump

If crashed_cleanup is set to *y*, this determines the number of memory dump files retained for later diagnosis.

stats_runningavg

Defines the time window in seconds for the calculation of running response time averages. The default is 600.

htmlencode

Specifies whether to encode HTML tags in data as plain text in order to prevent attacks by inserting executable code into data. Valid values are *y* and *n*. The default value is *n*.

proxy_server

Defines a proxy server, which acts as an intermediary on requests for resources from other servers.

proxy_port

Defines the port number of the proxy server.

proxy_user

Defines a user ID for the proxy server.

proxy_password

Defines a password for the user ID connecting to the proxy server.

outbound_ssl_certificate_file

Required for SSL when using self-signed certificates on i5 systems.

outbound_ssl_certificate_passphrase

Required for SSL when using self-signed certificates on i5 systems.

outbound_ssl_certificate_label

Required for SSL when using self-signed certificates on i5 systems.

outbound_ssl_certificate_keyfile

Required for SSL when using self-signed certificates on i5 systems.

outbound_ssl_auth_server

Required for SSL when using self-signed certificates on i5 systems.

seslog_max_lines

Defines the upper limit for the number of rows in a session log. When the limit is exceeded, only the latest records will be kept in the session log and the number of records will equal the limit. The default is 1000.

popup_timer

Defines the time in seconds to wait before the popup timer appears. The default is 5.

2. Make your changes, and click *Save*.

For details on the server configuration, see the *TIBCO WebFOCUS® Reporting Server Administration* manual.

Procedure: How to Set a Value for Global Variables

1. In the DMC, right-click the *Workspace* folder, point to *Settings*, then *FOCUS Sets and Info*, and then click *Core Engine Global Variables*.

   The Core Engine Global Variables dialog box opens.

2. Click *Add New*.
3. From the Command drop-down menu, select *DEFAULT* or *SET*.
4. In the Name box, enter the name for the global variable. Do not use ampersands.
5. In the Expression box, enter the value of the variable, enclosed in single quotes for character values.
6. Click *Save*.
7. Click *Cancel*.

Procedure: How to Create a Backup of Server Configuration Files

1. In the DMC, expand the server and then expand the *Workspace* folder.
2. Right-click the *Configuration Files* folder and click *New Backup*.

3. Click *OK* to confirm the backup.

Procedure: How to Restore TIBCO WebFOCUS Reporting Server Configuration Files

1. In the DMC, expand the server and then expand the *Workspace* folder.
2. Right-click the *Configuration Files* folder and click *Manage Backups*.

   The Configuration Files Backup dialog box opens.
3. Right-click the file you want to restore, or expand it and right-click on a specific file. Click *Restore*.

Procedure: How to Update Application Settings

Application settings control some aspects of use of application directories. You must be a server administrator to change any of these settings, and they require a restart of the server to take effect.

When you connect to the server, a temporary directory called foccache is added as the first directory in the search path. When you want to be able to reuse data within the same session, you can store the data in the form of a HOLD, SAVE, or SAVEB file in the foccache directory. As long as the session remains active, the files stored in the foccache directory can be referenced in requests. Foccache is also used to store some system created temporary files.

To change server application settings:

1. From the DMC, log on with a server administrator user ID.
2. Right-click the *Workspace* folder, point to *Settings*, then *Workspace SET*, and click *Application Settings*.

   The Application Settings dialog box opens.

   The following options are available:

   foccache_dir

   Defines the location of the foccache directory. The default location is edatemp/foccache/….

   foccache_maxage

   Defines the maximum number of minutes that a foccache directory will last. When the maximum foccache retention time is reached, the foccache directory is removed by the server and can no longer be referenced. The default is 180 minutes.

nested_app

When set to y, nested applications are enabled. This is the default value. To disable, set to n.

upload_allowed

Lists the file extensions that are allowed for upload files.

A home application directory provides each user with a private application directory with full access privileges to store private focexecs and data for the test runs and ad hoc reports.

homeapps

The operating system specific pathname. Defines the location of home application root directory. If defined, each connected user will see the *My Home* applications folder on the Application tree. User can create new applications (home applications) in this folder. These applications will be visible and available only to the connected user and the server administrators. The user will have full access to their home applications. For this setting to work, nested_app should be set to y.

3. Make any necessary changes and click *Save and Restart Server*.

Procedure: How to Make a Global Variable Persistent in a Session

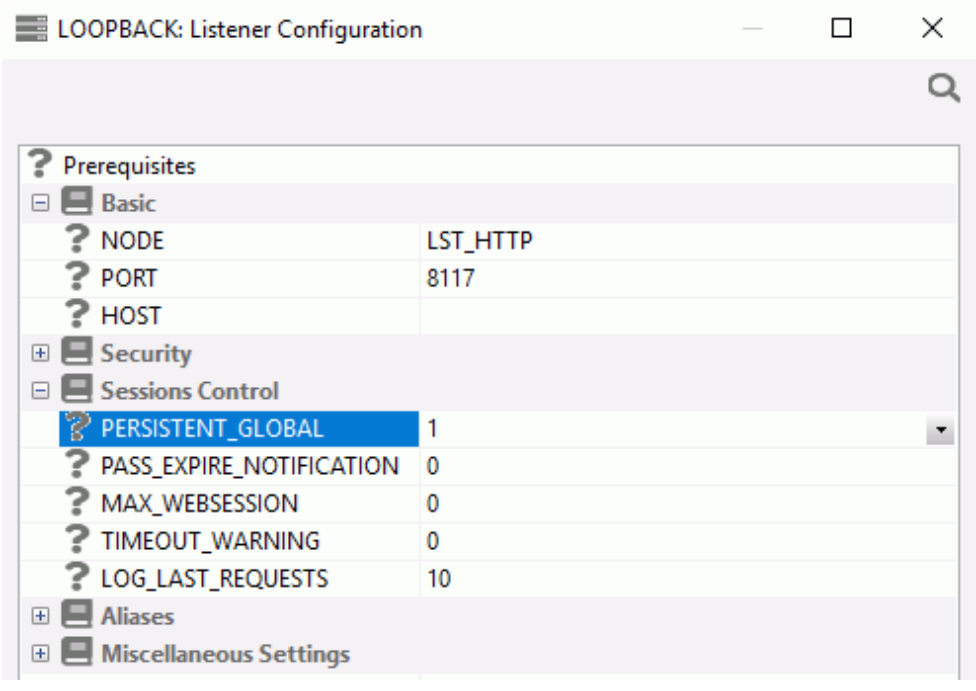
Global variables can be used by various users, regardless of the fact that each user has their own profile, by changing a listener configuration property.

Note: This is available for HTTP Listener only.

1. In the DMC, expand the *Workspace* folder.
2. Expand the *Special Services and Listeners* folder and right-click *TCP/HTTP*.
3. Click *Properties of HTTP*.

   The Listener Configuration dialog box opens.
4. Expand the *Sessions Control* section.

5. Select *1* in the PERSISTENT_GLOBAL drop-down menu, as shown in the following image.



6. Click *Save and Restart Server*.

The values of any global variables will now be kept active as long as you stay connected to the server.

Using Secure Shell FTP (SFTP)

Before using SFTP with Data Migrator, you need to set up SSH connections without passwords.

For example, if your Server is a on a Linux system called *local* and you want to retrieve or write files on a server called *remote*, you should log on to your local system with your user ID, for example, *user*. Then, you need to generate a key file:

```
/home/user/$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
10:7d:25:cb:ff:1d:c5:f4:1e:dc:7d:44:20:fd:21:56 home@local
/home/user/$
```

This process creates two files in your .ssh directory: your private key id_rsa and your public key id_rsa.pub. Then, you need to add your public key to the authorized users on the remote server. You can do this using FTP.

On most Linux systems you should able use the command ssh-copy-id.

For *remote*, enter the name of your remote (SFTP) server.

```
/home/user$ ssh-copy-id user@remote
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
     "/home/user/.ssh/id_rsa.pub"
```

Note: If the command ssh-copy-id is not available on your system, use the following command instead:

```
/home/user$ cat ~/.ssh/id_rsa.pub | ssh user@remote
    "cat >> ~/.ssh/authorized_keys"
user@remote's password: password
/home/user$
```

You can verify if this worked by using SFTP. You should not get prompted for a password.

```
/home/user/$ sftp user@remote
Connecting to remote...
sftp> quit
/home/user/$
```

Your connection is now set up. In the Data Management Console, expand the *Adapters* folder. Then expand the *Configured* folder. Right-click *Delimited File* (or *Fixed Format File*) and click *Add Connection*. You can now select *SFTP* as the transfer protocol.

Using SFTP From Windows

Starting in 2015 and the Windows 10 release, you can enable and use SSH and SFTP on the Windows platform.

In order to use SFTP with Data Migrator, you need to install OpenSSH and set up secured connections without a password from your Windows machine.

Note: Since you are using a local machine, it should be sufficient to only install a client SSH software.

Before following the procedures below, verify if you have SSH integrated on your Windows machine by opening the Command Prompt window and entering *ssh*.

If you get a response describing usage syntax, then you have SSH enabled on your Windows machine. Otherwise, please see *How to Install the SSH Client on Windows* on page 80.

To ensure that SFTP is integrated with your Windows machine after SSH is enabled, open the Command Prompt window and enter *sftp*.

Procedure: How to Install the SSH Client on Windows

To install the SSH client:

1. On your Windows machine, click the Windows Start menu.
2. Click *Settings* to open the Settings window.
3. Click *Apps* and then click *Apps & features*.
4. Click *Manage optional features* and then click *Add a feature*.
5. Scroll through the list of features and click *Open SSH Client* to expand the window, and then click *Install*.
6. After the installation is complete, open the System Properties for the machine and click *Environment Variables*.

   The Environment Variables dialog box opens.
7. In the System Variables section, verify that %SYSTEMROOT%\System32\OpenSSH\ was added to the Path.

   Note: You might need to reboot your machine to complete the installation process.

Procedure: How to Establish a Secure Password-less SFTP Connection

You can establish a SFTP connection that does not require a password by following these steps:

1. On your remote machine ensure that you have the *.ssh* sub-directory located in the */home/user* directory, and an *authorized_keys* file that will store a public key from your local machine. If the directory and the file do not exist, you need to create them. The following example syntax shows how you can create these files:

   ```
   /home/user/$ mkdir .ssh
   /home/user/$ touch .ssh/authorized_keys
   ```

2. On your local Windows machine ensure that you have the *.ssh* sub-directory in the */home/user* directory, for example, C:\Users\userid\.ssh. To create private and public keys (id_rsa and id_rsa.pub), enter the following command:

   ```
   /home/user/$ ssh-keygen -t rsa
   ```

   You can use all of the default values by pressing the Enter key at the prompts, or enter your preferred values.
3. Connect to your remote machine and append your public key (content of id_rsa.pub) to the authorized_keys file on the remote machine:

```
/home/user/$ ftp remuser@remotemachine
ftp> lcd .ssh
ftp> cd .ssh
ftp> append id_rsa.pub authorized_keys
  [ or /home/user/$ cat ~/.ssh/id_rsa.pub | (ssh remuser@remotemachine "cat >> ~/.ssh/
authorized_keys")  ]
ftp> quit
/home/user/$ sftp remuser@remotemachine
```

The last command line may ask you if you want to save the keys permanently. Enter *yes* when prompted.

4. Verify the SFTP connection. You should not get prompted for a password.

```
/home/user/$ sftp remuser@remotemachine
Connecting to remote...
sftp> quit
```

Using SFTP From Early Windows Versions

While Windows does not include an SFTP client program, there are programs available from both open source and commercial sources that implement the protocol.

One free source is PuTTY, which supplies a collection of utilities including a client program (psftp.exe) with similar functionality. They also provide puttygen.exe for generating private or public key pairs and converting an existing private key to a file in their .ppk format.

To use PuTTY's PSFTP:

1. Go to the PuTTY Download Page, download the MSI installer, and run it to install and configure their utilities. Alternatively, download *psftp.exe* and *puttygen.exe* to a directory in your PATH.
2. Download a private key to your Windows system or generate a new key using puttygen. Save it to a directory of your choice, for example, C:\putty\id_rsa.ppk.
3. If necessary, add your public key to ~/.ssh/authorized_keys on the remote server.
4. Test the connection from the command line using the following lines:

```
drive:\> psftp -v -i "drive:\putty\id_rsa.ppk" user@remote
Using username "user"
The server's host key is not cached in the registry
...
Store key in cache? (y/n) y
Remote working directory is /home/user
psftp> ls
psftp> quit
```

where:

drive
Is the drive on which the private key is saved.

user
Is your user ID.

remote
Is the name of your remote server.

You should not be prompted for a password or passphrase. However, if you have not previously connected to the server, you will be prompted to store the key in the cache. Reply *y* and you will not be prompted again. You must do this so that the Server does not receive this prompt.

5. In your PATH, create a file called sftp.bat with the following lines:

```
@echo off
psftp -i "drive:\putty\id_rsa.ppk" %1 %4 %5 %6 –batch
```

where:

drive
Is the drive on which the private key is saved.

If your Server is running, shut it down and then start it again. You can now configure a connection for a (delimited) file for a remote connection using SFTP.

Upgrading TIBCO Data Migrator From Release 4.3 or 5.1

To upgrade from predecessor products Copy Manager or ETL Manager, you must first upgrade to Release 5.2, 5.3, 7.1, or 7.6 following the instructions in the appropriate *TIBCO® Data Migrator User's Guide*.

Upgrading TIBCO Data Migrator From Release 5.2 and Higher

For most prior releases, upgrading to Release 7.7 or 82 always results in a new installation of the Server. This allows you to run your new and old server installations in parallel, if needed.

Note: This is not the case when replacing Release 7.6. For more information on upgrading from that release, see *Upgrading TIBCO Data Migrator from Release 7.6* on page 85.

This section lists some things to consider when upgrading Data Migrator. Refer to *Revised Functionality* on page 22 to review functional changes found in Release 7.7 or 82 to see if they affect your flows.

Port Number

Each Server listens on a unique series of port numbers for connections from the Data Management Console, a web browser, and other IP services. By default, port 8120 is used for TCP services and 8121 through 8123 are used for HTTP services.

To have multiple servers running on the same system, such as the 77 and 82 servers, they must each have their own unique set of port numbers.

Configuration Files

A new installation means that you will need to recreate or migrate your configuration information. Consider whether you want the new and old installations to use the same database connections. For instructions on how to migrate the configuration files, see *How to Migrate Configuration Information* on page 84.

Note: If you choose to migrate configuration information from an older release, you should do so before entering any new configuration settings. This prevents overriding existing configuration information by the migration process.

Profiles and Applications

If you have installed your new server installation on a different system from the prior release, you may also need to migrate your APPS directories where flows, procedures, and metadata are stored. You may also need to migrate your PROFILES directory, which is where individual user profiles are stored. This can be done by using operating system copy commands or from Explorer on Windows.

Data Flows

Data flows and procedures that ran in prior release should run in this release without modification. They will be automatically upgraded to take advantage of new functionality when they are opened and resaved from the DMC.

Scheduler

If you have scheduled flows, you should consider whether you want the scheduler on both the new and old releases to initiate scheduled flows from the same application directories.

Data Management Console

The Data Management Console (DMC) and Server releases should both be the same. When upgrading the Server, the new version of the DMC must also be installed for each user that connects to the server.

Note: Two or more installations of the DMC can be on one PC in order to connect to different server releases.

Procedure: How to Migrate Configuration Information

To migrate configuration information from a previous release:

1. In the navigation pane of the DMC, expand the server node and then right-click the *Workspace* folder.
2. Click *Migrate*.

   The Migrate dialog box opens.
3. Enter the full path of the configuration instance directory (EDACONF). On Windows, for example, this path is C:\ibi\srv76\dm.
4. Click *Migrate*.

Procedure: How to Migrate TIBCO Data Migrator From One System to Another

Note: If the new Server has been installed with security and data adapters have been configured, you do not need the configuration files from the previous environment.

1. Locate the following files:

   - ❏ edaserve.cfg
   - ❏ edaenv.cfg (If it exists)
   - ❏ edasprof.prf
2. Backup these files using the Server Web Console:

   a. Open the *Workspace* folder.

   b. Right-click the *Configuration Files* folder and click *New Backup*.

   c. Click *OK*.
3. Copy the following files into a directory path on the new environment so they can be used with the migration process. The paths used are examples for a Windows system:

   From the C:\ibi\srv82\dm\bin directory into the C:\ibi\cnfg\bin directory:

   - ❏ edaserve.cfg
   - ❏ edaenv.cfg (If it exists)

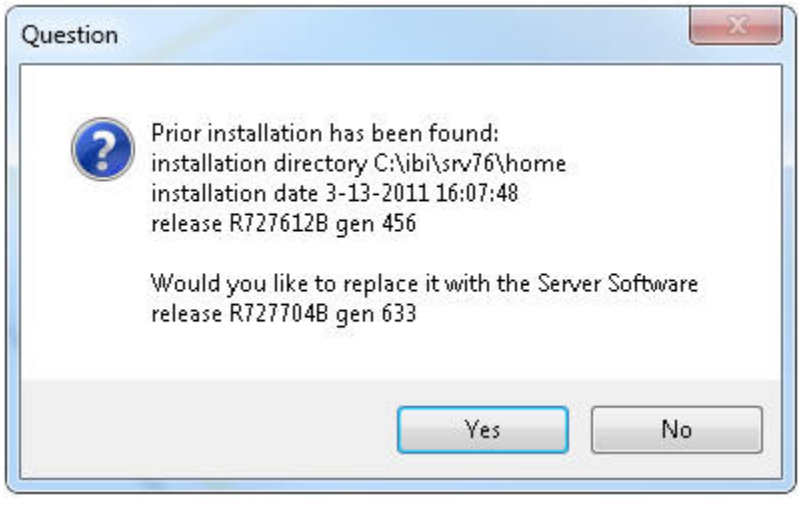
   From the C:\ibi\srv82\dm\etc directory into the C:\ibi\cnfg\etc directory:

   - ❏ edasprof.prf

4. After migrating the configuration files, start the Server.
5. Place the following files from the previous environment into the new environment. They should be manually copied and placed into the same directories in the new environment:
   - ❏ From the C:\ibi\profiles directory on the old machine, the admin.cfg file and any user profiles (.prf) files.
   - ❏ From the C:\ibi\apps directory on the old machine, all of the sub application directories.
6. Start the Server.

Upgrading TIBCO Data Migrator from Release 7.6

If you have a previous installation of Data Migrator Release 7.6 installed on your system, a message like the one below will appear that gives you the opportunity to replace your current installation with Release 77.



If you click *No*, then a new installation is performed, as described above. This the preferred option for most users, as it allows for parallel operation if needed.

If you click *Yes*, then your 7.6 installation is replaced with a 77 installation. You will then have a Release 77 installation in a directory identified (by default) as SRV76, but you will not be able to use Release 7.6 unless you re-install it.

In this case, there is no need to migrate your configuration files. However, the Data Migrator Log and Statistics table formats have changed and they must be re-created, as explained in *How to Recreate the Log and Statistics Tables* on page 86.

Upgrading TIBCO Data Migrator to Release 7.7.07 or 82

In this release, the length of the application path names have been increased from 64 to 256 characters. This requires that the Data Migrator Log and Statistics tables are recreated. In addition, you should clear the user preferences.

Procedure: How to Recreate the Log and Statistics Tables

You can recreate the Data Migrator Log and Statistics tables using a Server Administrator user ID from the Data Management Console or the Web Console, by following these steps:

1. Click the *Workspace* folder or tab.
2. Expand the *Logs and Traces* folder.
3. Right-click *Data Migrator Log and Statistics* and click *Recreate*.
4. Click *OK* in the Confirm dialog box.

Procedure: How to Clear User Preferences

After the upgrade, the recommended best practice is to clear the user preferences by following these steps:

1. In the *Tools* section on the ribbon, click *Options*, then *General*.
2. Select *Clear User Preferences*.
3. Restart the Data Management Console.

Chapter 3

Planning a TIBCO Data Migrator Application

Although Data Migrator facilitates all aspects of your design task, experience has convinced us that the results of your efforts will be more rewarding if you do a little advanced planning.

This chapter provides important background information that will help you make optimal decisions to fulfill data management needs of your site.

In this chapter:

- Designing Your Application
- Using Synonyms for Data Sources and Data Targets
- Reviewing Your Data Sources
- Reviewing Your Data Targets
- Establishing Security

Designing Your Application

Data warehouses and/or data marts are created for a variety of reasons. Knowing the factors leading to the decision for extracting or moving data will help you identify the issues for DM management that will have the greatest impact on your success. The ultimate goal is always to have an automated process that moves and converts all of your data in a very small period of time. However, it helps to know what issues are most important in your company's criteria for a successful data warehouse. Are you consuming large amounts of time in the management of 3GL extraction programs and load routines? Are you uncomfortable with ad hoc user access to production legacy systems? Is your network unable to handle the performance requirements for decision support and analysis?

If you consider the reasons for warehousing, your attention will be properly focused on the critical variables. Data Migrator provides you with the flexibility to control the variables inherent at your site and in your data.

From an application perspective, you must evaluate what will direct your overall data movement strategy. Considering the following questions will help you learn more about the capabilities of Data Migrator and how to control them.

- **What is the typical volume of data being migrated?** One million bytes, 100 million, or several gigabytes of data?
- **How often will data be migrated?** Once every night or every 15 minutes? A single large load once per month?
- **What time of day or night is optimal for moving the data?** Some sites cannot get access to data until after the nightly production jobs are complete (which may be at 5:30 AM). When is the warehouse required? Are different time zones involved? This complex question impacts not only your performance requirements, but also your scheduling methodology.
- **Is there a key value in the source data that will identify changed rows?** Data Migrator can locate changed data using an SQL WHERE criteria, unless an extract file exists with delta changes. Is this file available? Do you know what the WHERE criteria should look like? Using *Change Data Capture*, an optional, add-on component available for selected relational databases, you can load a data target with only the records that have changed.
- **How many designers will be developing extractions with Data Migrator?** Some sites designate a single Data Migrator administrator who is responsible for all the extract and load routines that are used to develop their data marts. (There are rarely any conflicts in such an environment; one user has complete control over the requests, the source data, and the target locations that will be written.) Other sites have a team of developers interacting to create a collective data warehouse.

Using Synonyms for Data Sources and Data Targets

Servers, including the Reporting Server, access local and remote data sources and data targets using synonyms, which contain the metadata for a data source. This metadata consists of the attributes that describe the data source or target.

Since DM processing depends upon the appropriate metadata being available to the Server and any remote servers in your configuration, you must create synonyms for all data sources and any existing data targets before you can use them in a flow. For information about providing metadata to servers, see *Setup At a Glance* on page 28 and *Remote TIBCO WebFOCUS Reporting Server Setup At a Glance* on page 30.

The Data Management Console application directory tree lists the synonyms available to the Data Migrator user for data sources and data targets. This structure is also known as the users Application Path on the server.

Reference: **General Guidelines for Synonyms**

- The first character in a synonym name must be a letter. The remaining characters can be any combination of letters, numbers, and underscores (_). The synonym name cannot exceed 64 characters. When naming a synonym, you must avoid the following:
  - Names containing capital letters.
  - A name beginning with the letters ETL or SYS. These prefixes are reserved for Data Migrator internal use.
  - In a FOCUS/FDS or XFOCUS data source, a non-unique field name.
  - Names that are used as commands within Data Migrator. These are HOLD, SQLIN, SQLGET, and MODIFY.
  - Names containing spaces, dashes, and other non-standard characters in both the source and target data columns (*, &, #, $, %, @).
- Synonyms must not exceed any of the following:
  - Approximately 3,000 columns.
  - 30K of total column name character length.
  - 16K for single LRECL (total length of field formats, one instance of a row).
  - Column names cannot exceed 128 characters.

  See the *TIBCO WebFOCUS® Adapter Administration* manual for suggestions on how to access your structures without pressing these limits.
- For column names, avoid the following:
  - Special characters. Only alphanumeric characters and underscores are permitted in column names.
  - Names beginning with the capital letter E followed by a two or three digit number, for example, E01 - E999.
  - Reserved words, including:

ALIAS, ALL, ALTER, AND, ANY, AS, ASC, AUTHORIZATION, AVG, BEGIN, BETWEEN, BINARY, BIT, BOTH, BY, CALL, CASE, CAST, CHAR, CHARACTER, CHECK, CLOSE, COALESCE, COMMIT, CONNECT, CORRESPONDING, COUNT, CREATE, CROSS, CURRENT, CURSOR, DATE, DEALLOCATE, DEC, DECIMAL, DECLARE, DEFAULT, DELETE, DESC, DISTINCT, DO, DOUBLE, DROP, ELSE, ELSEIF, END, ESCAPE, EXCEPT, EXECUTE, EXISTS, FETCH, FLOAT, FOR, FOREIGN, FROM, FULL, FUNCTION, GET, GRANT, GROUP, HAVING, IF, IN, INDEX, *INDICATOR, INNER, INOUT, INSERT, INT, INTEGER, INTERSECT, INTO, IS, ITERATE, JOIN, KEY, LEADING, LEAVE, LEFT, LIKE, LOOP, MAX, MIN, NATURAL, NOT, NULL, NULLIF, NUMERIC, OF, ON, ONLY, OPEN, OPTION, OR, ORDER, OUT, OUTER, PRECISION, PREPARE, PRIMARY, PROCEDURE, REAL, REFERENCES, REPEAT, RETURN, RETURNS, REVOKE, RIGHT, ROLLBACK, ROW, ROWS, SCHEMA, SELECT, SET, SMALLINT, SOME, SUM, TABLE, THEN, TIME, TIMESTAMP, TO, TRAILING, UNION, UNIQUE, UNTIL, UPDATE, USER, USING, VALUES, VARCHAR, VARYING, VIEW, WHEN, WHERE, WHILE, WITH, WORK, END, COLFETCH, CONCAT, DATABASE, DATETIME, DAY, DAYS, EXPLAIN, GRAPHIC, HOUR, HOURS, IMAGE, INCLUDE, INDEX, LOCK, LOGICAL, LONG, MICROSECOND, MICROSECONDS, MILLISECOND, MILLISECONDS, MINUTE, MINUTES, MONEY, MONTH, MONTHS, NUMBER, OPTIMIZE, PACKAGE, PERCENT, PLAN, PROGRAM, PURGE, QUERYNO, RAW, RESET, SECOND, SECONDS, SERIAL, SMALLFLOAT, STOGROUP, SYNONYM, SYSNAME, TABLESPACE, TEXT, TRUNCATE, USER_TYPE_NAME, VARBINARY, VARGRAPHIC, YEAR, YEARS

- Synonyms created using the Dynamic Columns option (NOCOLS parameter) should not be used with Data Migrator because their columns and formats will not display when building flows in the DMC.

Reference: Inclusion of Application Directory Names With Flow Components

The Reporting Server supports inclusion of application directory names in flow component labels. This means that data and process flows can store information about the application directories where the synonyms or procedures that they use are located.

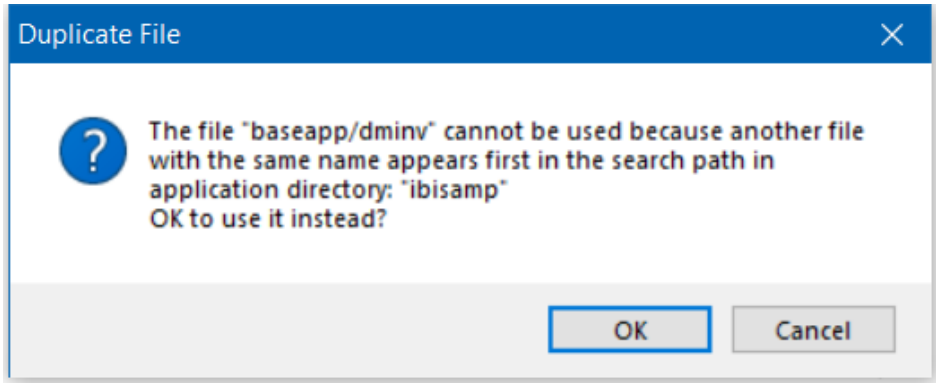
Prior versions of Data Migrator used application directories to group synonyms and requests (flows) into logical groups, and by default did not provide support for duplicate or multiple synonyms or requests with the same name, even if they were in different application directories.

For example, if you had two synonyms or requests with the same name and your application path was

```
ibisamp
demo
baseapp
```

it would not matter which synonym or request you selected. You would get the one from the application directory that appeared first in the application path, since the earlier version of the server did not support inclusion of the application directory. If you had synonyms or requests with the same name in more than one application directory, there would be no way to distinguish them, and the server would always use the one in the first application directory in the application path. Some users take advantage of this capability by constructing a run-time application path to control which synonym or flow is actually used.

For upward compatibility, when you build data flows, by default, they will not include application directory names. That means that the flow will not save the application directory name with the source or existing target. If you use a synonym that exists in more than one application directory, you will get a warning, which is shown in the following image.



The flow will pick up the source from the first application directory in the application path.

To include application directory names in data flows, on the Home tab, in the Tools groups, click *Options*. Click *Data Flow* and select the *Use application directory name with flow components* check box from the Data Flow Designer page. This way your flow will use exactly the synonym you specify.

For process flows, the default is to include application directory names.

Reference: Guidelines for Variable Length Character and Text Columns in Synonyms

If you are using variable length character or text columns in your synonyms, you should be aware of the following special considerations.

Variable Length Character Columns

Data Migrator supports variable length character columns that are described in a relational database as VARCHAR. This attribute enables you to copy such columns without padding them with spaces to the full column width or removing trailing spaces. It enables you, for example, to copy a column containing a single space.

This support is enabled by default.

When new synonyms are created, they are created with variable length columns described as

A*n*V

where:

n

Is the maximum column width.

Note: For fixed format data targets, the length of the value precedes the data value itself.

TEXT (TX) Columns

Data Migrator also supports columns described with ACTUAL=TX for Text columns. This includes data types MS SQL Server® VARCHAR(MAX) and Text, ORACLE LONG, and Db2® LONG VARCHAR. Sample Data displays only the first 50 characters of these columns.

Note: Columns described as TX cannot be used as a key column.

Reviewing Your Data Sources

Aspects of your data sources contribute to the performance of Data Migrator. Consider the following issues when reviewing your data sources:

- **Number of columns in an extract.** Data Migrator supports about 400 columns in an SQL command. Previous servers, however, support fewer. If more columns are required, you may need to have multiple flows or concatenate columns.
- **Number of tables.** The Server supports joining of up to 63 data sources, but the complexity increases as the number of files increases. Thorough knowledge of the data is required to determine if a multi-table join makes sense in any given environment. As a general rule, try to keep the number of tables to a minimum.

- **Location of your tables.** Is a cross-platform join required (for example, an z/OS® table to a UNIX® table)? Is all the data in one data source on one platform? Perhaps a cross-DBMS join (for example, C-ISAM® to Oracle) is needed? If so, this may impact performance.
- **Data types.** What data is being moved to the warehouse? Are all the data types supported by the current server environment? Also, be sure that the data adapter supports the ability to read and extract the data type required by your warehouse. Be sure that the joins you require can be supported by the data types of your columns. Consider the scale (number of decimal points) on floating-point columns, or site-specific columns that are often found in old data sources, such as company-specific date fields that require the use of a home-grown assembler routine.

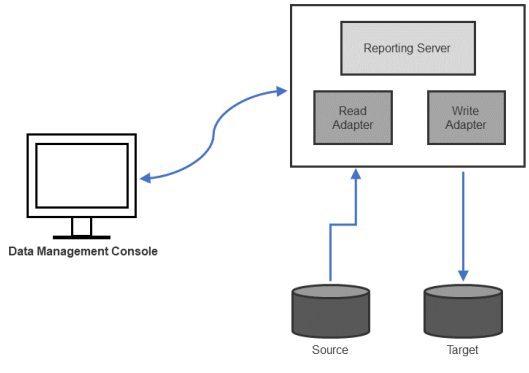
Accessing Data Sources

You can use Data Migrator to access data sources across an enterprise and move data from those data sources into various data targets. The Server accesses any WebFOCUS-supported data source, which includes numerous formats on over 35 platforms.

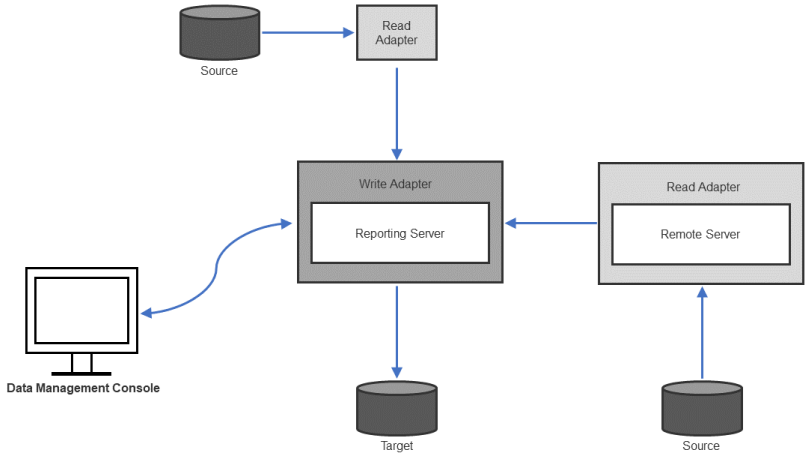
The Server can access data that resides locally, that is, on the Server itself. This is the simplest configuration, but not necessarily the most typical.

If all or some of your source data is located on one or more remote servers, the Server establishes a connection with the remote server(s) to access this data.

The following image illustrates how data can be moved or extracted to the Server from a source that is local to the Server.



The next image illustrates how data can be moved or extracted to a data target on the Server from a data source that is local to the Server, as well as one that is located on a remote server. With this configuration, despite the different locations of the data sources, the Reporting Server can join them together and access the joined files.



Reviewing Your Data Targets

The data targets you use for your application affect the performance of Data Migrator. Consider the following issues that pertain to the tables or files being written by Data Migrator:

- **Row processing logic.** Is this a total refresh? Will the request be deleting all rows or dropping the table before starting? Is an update or delete necessary? The processing logic is specified in your data flows.
- **Location of the tables.** Are the tables local to the machine where the server will be running? What connection information will be needed to resolve the names for remote data sources? Can it be handled in the Access File for the synonym? The DMC allows you to specify these items, but in many cases it is best to allow the Hub Server to manage the location of tables. Review the server profile (EDASPROF.PRF) or the profile for your user ID, as well as the Access File that is used with the Server to establish table names and their storage locations.

Data Migrator extracts the data from a data source and loads it into a data target, which may be:

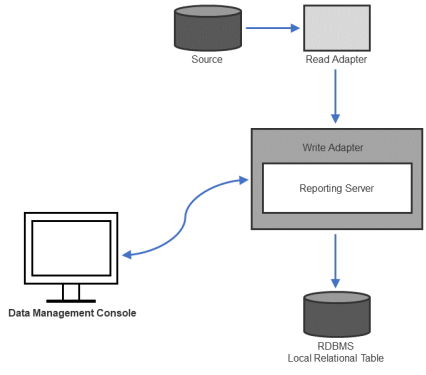
- A relational data target. For details, see *Relational Data Targets* on page 95.

- A non-relational data target. For details, see *Non-Relational Data Targets* on page 96.
- Remote destination. For details, see *Remote Target Destinations* on page 96.

Relational Data Targets

Depending on the platform, supported relational data targets currently include the following: Access, Db2, Informix, MySQL, Nucleus, ODBC, Oracle, SQL Server, Sybase, and Teradata.

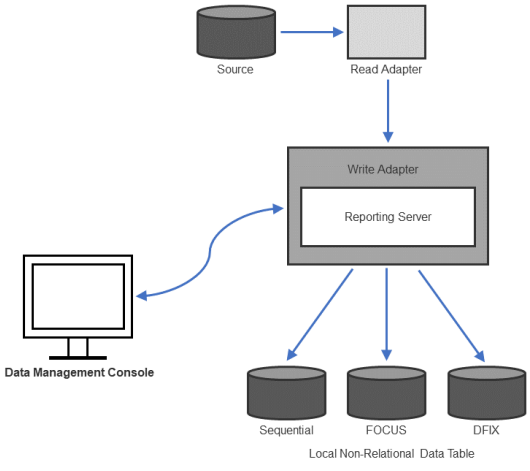
In the following diagram, Data Migrator creates a target table or writes to an existing target table in a local RDBMS which resides on the Server (the recommended configuration for a data target).



Note: EDA is only supported for existing targets. The name of the remote server must match the real server name of the subserver. This option should only be used for small data volumes. For large volumes, you should install the server on the platform where the data will be loaded.

Non-Relational Data Targets

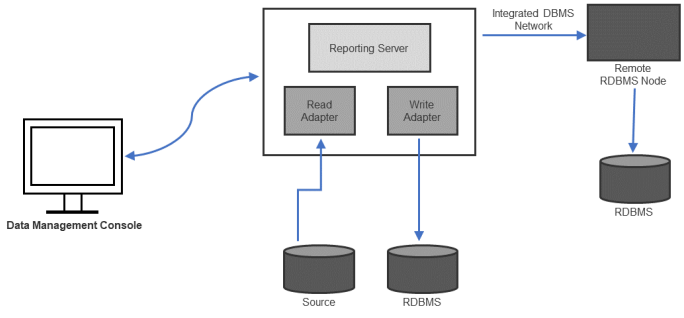
You can create non-relational data targets of the following types on the Server (the recommended configuration for a data target): a FOCUS/FDS or XFOCUS data target, or a fixed format sequential file of different formats. A fixed format sequential file is created in the character set appropriate for the server platform (for example, ASCII on UNIX or EBCDIC on z/OS). In the following diagram, Data Migrator creates or writes to an existing non-relational data target.



Remote Target Destinations

Many relational data source management systems provide for remote table access. If properly configured, the DM server supports the syntax necessary to write to remote targets (for example, SQL*NET provides access to remote Oracle tables).

The following figure illustrates how a DBMS located on the Server uses networking capabilities to write to a remote data target.



Reference: General Guidelines for File and Application Directory Names

The following guidelines describe server support for file and application directory names.

- ❏ All file and application directory names must be lowercase.
- ❏ All file names must include an extension.

Reference: General Guidelines for Data Targets

- ❏ Data Migrator supports up to 64 characters for the synonym name used with a data target, or the limit set by the operating system or DBMS. The 64-character limit is for the total name, including data source and owner details.

  For example, if you are loading a relational data target, the full name may exceed 64 characters since you can add data source and owner details to the synonym name.

- ❏ Avoid the following when naming data targets:
  - ❏ Names beginning with the capital letter E followed by a two or three digit number (for example, E01 - E999).
  - ❏ Names beginning with the capital letter T followed by a number (for example, T1 - T16).

- When you write or move data to an existing relational target, ensure that the correct access rights are assigned to the approved group of designers. New relational targets generally default to assigning an owner ID equal to the logon ID, although this may vary based on the type of data source selected.
- Data Migrator is often used to create data targets in the local relational subsystem. This requires that the user ID processing flows has appropriate access to the target RDBMS.
  - If the data target is a new table, the user ID must have permission to issue CREATE TABLE statements.
  - If the data target is an existing table, transaction update access is necessary.
- Relational data targets created in Data Migrator are owned by the user ID specified in the connection string in the server profile.
- Fixed format, FOCUS/FDS, and XFOCUS file data target attributes are controlled independently by the designer. Therefore, they may only create conflict if users select the same file name (operating system dependent) for their output. This can be avoided by establishing a convention for where users will write their extracted answer sets.
- Where fixed format file data targets are being created, the user ID of the processing request must have access to the subdirectory or data set that is currently allocated to receive the answer set. Fixed format files created in Data Migrator are owned by the server user ID.

Note: For the best throughput loading relational tables, the Reporting server should be on the same system as the RDBMS server. Throughput to a remote server is dependent on your network speed and traffic.

Establishing Security

If only a single Data Migrator user will be designing flows, no alterations to the Server configuration are necessary. However, if the Server supports multiple users, the server administrator may need to establish separate user IDs and profiles to control access to DM application directories.

Note: Establishing security for TIBCO Servers, including the Reporting Server on z/OS requires additional consideration. For details, refer to *Step 7. Configure Server Security* in Chapter 4, *Installation for z/OS*, in the *TIBCO WebFOCUS® Reporting Server Installation* manual.

Running Flows

The agent created when you connect to a Reporting Server has an associated logon user ID. Local file, directory, and resource security is controlled by that user ID.

For scheduled flows, the DM components that a user ID can see and run from its Application Path are controlled from *sched_run_id* option, accessible using the Type setting in the New Tenant dialog box, accessed from the Manage Scheduler Userids dialog box.

When *sched_run_id* is set to:

- **server_admin_id.** Scheduled DM jobs are run under the first ID that appears in the list of server administrators displayed on the Access Control page. If the ID does not have a password specified in the Access Control tab, a profile for that user ID must be created. server_admin_id is the default.
- **user.** Scheduled DM jobs are run under the user ID that was used to save the flow. The Application Path specified in the user profile is utilized.
- **RunID.** Scheduled DM jobs will run under the specified user ID. If you select this option, an additional drop-down menu appears where you can select a user ID from a list of configured user IDs.

If security is ON and you set sched_run_id to user to run a scheduled flow for a certain user ID:

1. The user ID must be a valid user on the system.
2. The user ID must be set to an access level of either SERVER or APPLICATION from the Access Control page.
3. The password for the user must be set. A SERVER-level administrator can set the password for a SERVER-level ID from the Access Control page when adding a user.

If a Data Migrator user with an APPLICATION-level ID wants to run scheduler requests:

1. A SERVER administrator must make them an APP administrator from the Access Control page.
2. The user must set their password on the User Information page.
3. The scheduler must be restarted. (Restarting the server will also restart the scheduler.)

For more information, see *Scheduler Configuration Dialog Box* on page 66.

Restricting the Application Paths Available to a User

By default, the server profile (EDASPROF.PRF) is run for all users when they connect to the Server to provide access to all application directories in the servers search path. However, an administrator can control a user's access to application directories by creating individual user profiles. Each user can then:

- Access only the application directories specified in the Application Path specified for that profile.

- Use synonyms in the specified Application Path.

For details, see *Authorizing TIBCO WebFOCUS Reporting Server Usage and Administration* on page 38.

It follows that the user ID that a flow runs under determines the user profile that is run. The profile controls the application directories available to the flow, as well as access to relational databases or source servers.

- If there is a profile associated with the user ID, then it is used.
- If there is no profile, then EDASPROF is used instead.

The user can only access the application directories defined in the profile being used.

You can set the Application Path from the DMC or the Web Console. For information on setting the Application Path from the DMC, see *Managing Application Directories and Configuring the Application Path* on page 41. For information on setting the Application Path from the Web Console, see the *TIBCO WebFOCUS® Reporting Server Administration* manual or the Web Console online help.

Running Scheduled Flows Under a User ID

By default, scheduled flows are run using the server admin ID.

To run all scheduled flows under the user ID that saved them, you need to:

1. Define a scheduler tenant and change the sched_scan_id and scan_run_id.
2. Create a new user (if the user ID does not already exist). This procedure will depend on your operating system.
3. Add users who can run flows as an Application Administrator.
4. Have the new users change their security settings.
5. Connect to the server as the new user in the DMC, schedule a flow, and save it.

Procedure: How to Define a Scheduler Tenant and Change the sched_scan_id and scan_run_id

1. Connect to the server with a Server Administration user ID.
2. In the navigation pane, expand the server, followed by the *Workspace* folder.
3. Expand the *Special Services and Listeners* folder.

   Note: If there is a *Start* option, the scheduler is not running. To run the scheduler, click *Start*.
4. Right-click *SCHEDULER* and click *Manage Tenants*.

The Manage Scheduler Userids dialog box opens.

5. Click *New*.

   The New Tenant .

6. To set the sched_scan_id for the tenant, select a user ID from the ScanID drop-down menu, which lists the configured user IDs from admin.cfg.
7. To set the sched_run_id for the tenant, select one of the following options from the Type drop-down menu:

   - Select *server_admin_id* to run all jobs as the Server Admin ID, which is the first user ID in admin.cfg.
   - Select *user* to run each job as the user that saved the flow.
   - Select *RunID* to specify a user ID for all jobs. If you select this option, an additional drop-down menu appears where you can select a user ID from a list of configured user IDs.

8. Click *OK* to configure the tenant.
9. Click *Save and Restart Scheduler*.

Note: It is recommended that each ScanID has its own Application Path set in their user profile. The same application directory should not appear in the application path of more than one ScanID.

Procedure: How to Add the New User as an Application Administrator

Note: If you want to run all scheduled flows under a user ID that does not already exist, you must create one using an operating system-specific procedure.

1. In the DMC, expand a server and then expand the *Access Control* folder.
2. Expand the *Roles* folder, right-click *Application Administrator*, and click *Register PTH <internal> User*.

   The User Registration dialog box opens.

3. In the User ID box, type the new user name.
4. Optionally, type a description and the email address of the user.
5. Click *Next*.

   The User Registration dialog box opens.

6. From the Inherit Privileges from drop-down menu, select *Application Administrator*.
7. Click *Register*.

8. Click *OK* to save your changes and register as a new user.

Procedure: How to Change a Password for Running Scheduled Flows

1. Log in to the DMC with an Administrator user ID.
2. In the navigation pane, expand the server and then the *Access Control* folder.
3. Expand the *Roles* folder and then expand the folder of the desired Role.
4. Right-click the user ID you want to manage and click *Properties*.
5. In the Optional password for scheduled runs section, type the new password, and re-type it to confirm the password.
6. Click *Update*.

Procedure: How to Connect to the TIBCO WebFOCUS Reporting Server as a New User and Schedule a Flow

1. In the DMC, right-click the server and click *Properties*.
2. Change the *User ID* and *Password* boxes in the Security section to the new values and click *Save*.
3. Disconnect and reconnect the server.
4. Open a process flow in the DMC and add a Schedule to the Start object.
5. Save the flow.

The Scheduled Events report will now list scheduled flows by the user ID that saved them.

Chapter 4

Working in the Data Management Console

This topic describes the Data Management Console interface and some key tasks you will perform as you design your flows.

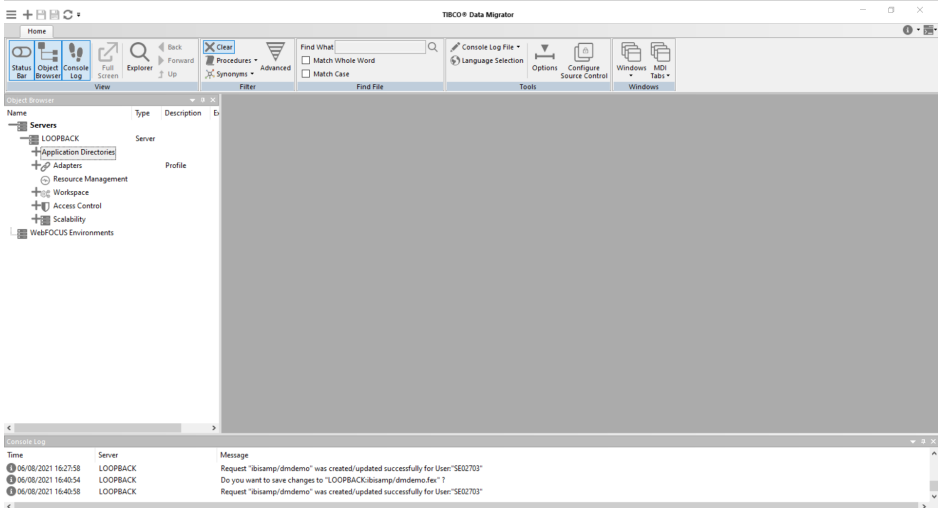
In this chapter:

- Data Management Console Interface
- Navigation Pane
- Workspace
- Text Editor
- Aerial View
- Ribbon Tabs
- Metadata Tab
- Text Editor Tab
- Console Log
- Setting Flow Properties
- Viewing Properties
- Setting User Preferences
- User Privileges
- Configuring National Language Support
- Ending a Data Management Console Session

Data Management Console Interface

The Data Management Console integrates all of the major tasks required to build flows. You can build flows, access the Web Console, and create, manipulate, and access the metadata associated with those flows. You can also create and run stored procedures.

The DMC interface has a navigation pane on the left, an area that will become the Data Migrator workspace on the right, and the Console Log on the bottom.



The interface contains:

- Servers that can be accessed by the DMC. The server application directories appear when you are logged on to a server. Until you connect to a server, only the servers are displayed.
- Application Directories listing the directories in the server path, which contain flows, procedures, synonyms, reports, adapters, and administrative functions.
- Application Directory Folders that contain flows, synonyms, procedures, and reports.
- Adapters and Workspace folders appear if you are logged on with a Server Administrator user ID. Monitor folder appears if you are logged on with an Application Administrator ID.
- The Console Log that displays the console log, including, but not limited to, messages when you connect and disconnect from the server, or execute flows and stored procedures.
- The ribbon that represents the most commonly performed activities in the DMC. The ribbon tab changes to reflect the currently open object.
- The Home tab in the ribbon, which lets you select which server objects are displayed, and which flows, procedures, or synonyms to use. The tab also lets you select explorer view when displaying information.
- The workspace where flows are built, synonyms are edited, reports are viewed, and the explorer view is shown.

Application Menu

In the upper-left corner of the Data Management Console, click the Menu button  to open the Application menu.

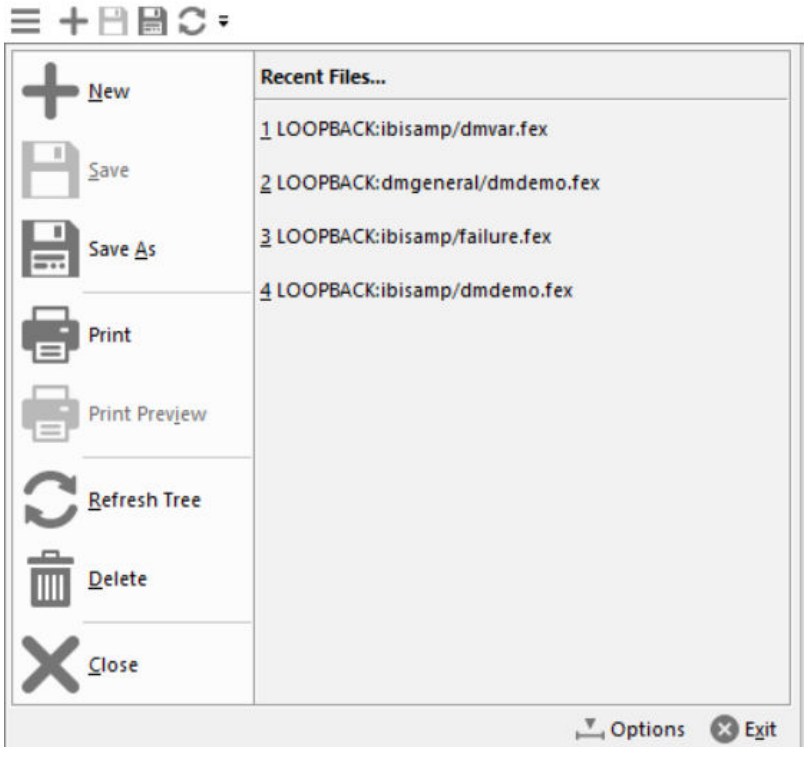
From this menu, you can:

- Create a new object.
- Save an active document.
- Save an active document under a new name.
- Preview or print an active document.
- Refresh the explorer tree.
- Delete a directory.
- Close an active document.
- Access the Options dialog box where you can set user preferences.
- Close an active document and exit the DMC.

Note: In the Application menu, the terms document and active document represent all files that you can create in the Data Management Console.

Tip: Double-click the Application button to exit the DMC.

The following image shows the Application menu.



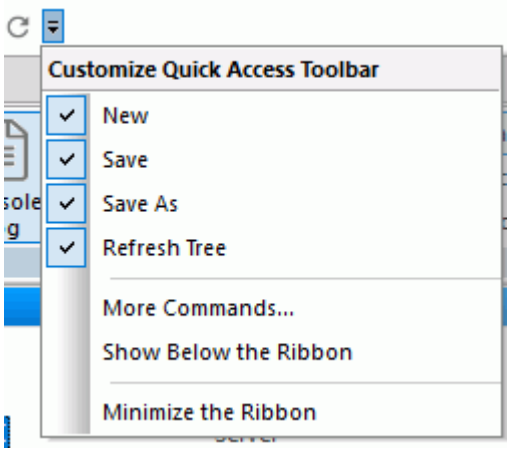
Quick Access Toolbar

The Quick Access Toolbar provides access to the most commonly used functions. It is located to the right of the Application menu button and is always visible no matter which options are selected.

The Quick Access Toolbar contains the following buttons, as default. Other buttons can be added as required.

- **New.** Creates new flows, stored procedures, synonyms, specialized flows, or server connections, as well as configuring adapters.
- **Save.** Saves the current procedure under a specified file name
- **Save As.** Saves the current procedure under a specified file name.
- **Refresh.** Refreshes the Explorer tree in the navigation pane.

On the Quick Access Toolbar, click the arrow to open the Customize Quick Access Toolbar menu, as shown in the following image.



From this menu, you can click:

- The check marks next to the New, Save, Save As, and Refresh options to add or remove these buttons from the toolbar.
- *More Commands* to select and add new commands to the toolbar.
- *Show Below the Ribbon* or *Show Above the Ribbon* to move the toolbar above or below the ribbon.
- *Minimize the Ribbon* to hide the ribbon.

Note: You can right-click the ribbon, groups, or commands on the ribbon, to access a modified version of the Customize Quick Access Toolbar menu. This menu contains the Add to Quick Access Toolbar, Customize Quick Access Toolbar, Show Quick Access Toolbar Below the Ribbon, and Minimize the Ribbon options.

Ribbon

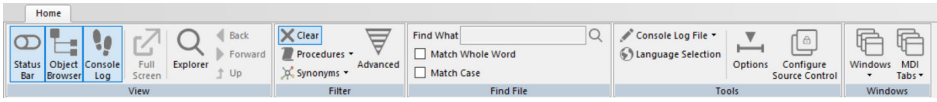
The DMC ribbon replaces traditional toolbars with a single, rectangular region that spans the width of the application window.

From the ribbon, you can:

- Select DMC commands to build and customize applications.
- Customize the appearance of the DMC interface from the Style menu.

- Open the Help menu to view the DMC technical documentation and online Help.

The following image shows the default ribbon in the Data Management Console.



Ribbon Organization

The DMC ribbon is composed of tabs. Each tab represents a subset of controls and commands that are organized into related groups depending on function or relationship. All tabs are named, all groups are labeled, and all icons represent a control or command.

Ribbon Tabs

When you launch the DMC for the first time, only the Home tab is visible. The Home tab is the only static tab and does not close or reopen as you work within the console. All other tabs on the ribbon are contextual, and will only open or close as you develop using the associated features.

The DMC tabs include:

- **Flow.** Opens when you are working with a data or process flow.
- **Direct Load Flow.** Opens when you are working with a Direct Load flow.
- **Text Editor.** Opens when you are editing files as text.
- **Metadata.** Opens when you are working with a synonym in the Synonym Editor.
- **Report.** Opens when you are viewing a report.

Style Menu

In the upper-right corner of the ribbon, click the *Style* menu.

Select a classic theme to customize the Data Management Console. The themes available to you are determined by the themes you have installed for Microsoft® Office. The Style menu is always available.

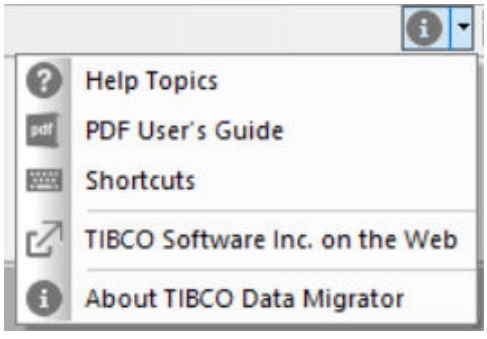
Note: When you close the DMC, the theme you selected is saved, and appears the next time you launch the product.

About Button

In the upper-right corner of the ribbon, click the *About* button.

Select *Help Topics* to access the online Help system or click *About* to view the product licensing information. The Help button is always available.

The following image shows the About button.



Note: You can also press the F1 button on your keyboard to access the online Help system.

Shortcut Keys for Navigating the Workspace

Shortcut keys enable you to move around the DMC interface using key strokes instead of a mouse. If your keyboard shortcuts are not underlined in the DMC, you can change the settings on your Windows system to enable the underlining of such keys by performing the following steps:

1. From your Windows Control Panel, click *Ease of Access Center*.
2. Click *Make the keyboard easier to use*.
3. Select the *Underline keyboard shortcuts and access keys* check box.
4. Click *Apply*.
5. Click *OK* to close the window.

The following combinations are available:

Key Combination	Result
Alt + 0	Moves focus to the navigation pane.
Alt + 1	Opens the New dialog box.
Alt + 2	Opens the Save dialog box.
Alt + 3	Performs the action of the 3rd item on the Quick Access Toolbar.

Key Combination	Result
Alt + 4	Performs the action of the 4th item on the Quick Access Toolbar.
Alt + 8	Shows or hides the browser.
Alt + 9	Shows or hides the Console Log.
Alt + U	Toggles full screen mode.
Shift + F10	Opens the right-click menu.
Menu key	Opens the right-click menu.
Ctrl + 2	Shows or hides the status bar.
Ctrl + E	Shows or hides explorer view.

Navigation Pane

The navigation pane on the left side of the interface contains servers, which in turn contain application directories. The application directories contain flows, procedures, reports, synonyms, adapters, and administrative functions.

The navigation pane has three or more top-level folders, depending on your administration level:

- **Application Directories folder.** Provides access to flows, procedures, and reports that can be generated from a flow. It also provides access to synonyms that can be used as data sources and data targets. For details. see *Procedures Shortcut Menu Options* on page 114.
- **Adapters folder.** Allows you to add and configure adapters. This folder is only available if you are logged in as an Application Administrator.

  For details, see *Configuring Data Adapters* on page 48.
- **Resource Management.** Allows you to configure Resource Management.
- **Workspace folder.** Allows you to configure and monitor the server. This folder is only available if you are logged in as a Server Administrator.
- **Access Control.** Allows you to control what users can access the server and their access level. This folder is only available if you are logged in as a Server Administrator.

- **Scalability.** Used for performance testing.

Reference: Server Shortcut Menu Options

Server shortcut menu options are available if you right-click a server. The following options are available:

- **Connect/Disconnect.** Connects or disconnects the DMC to or from the server.
- **Start/Stop Server.** Starts or stops the server from running. This option is only for a local LOOPBACK server.
- **Alternate Authentication.** This option is only available if a third party cookie is used, such as Netweaver.
- **Access Web Console.** Launches the Web Console Applications page.
- **Other Options.** The following options are available:

  My Console:

  - **My General Privileges.** Shows the user privileges associated with the selected server.

    Note: This option is only available when connecting to a server with security enabled.
  - **Manage My Agents.** Runs the Data Services Agents report that shows all currently active agents.
  - **Save My Last Report.** Allows you to save a focexec with contents USERINFO.
  - **Edit My profile.** Allows you to edit your server profile.

    Note: This option is only available for a Server Administrator or when connecting to a server with security disabled.
  - **Login Info.** Displays your current log in status including your user ID, profile, and security role.
  - **Change Password.** Allows you to change your password on the server.

    Note: This option is only available when connecting to a server with security enabled.
  - **Stop My Requests.** Stops all Web session requests for the server.
  - **Image Directory.** Shows catalog of icons used in the Data Management Console and Web Console.

- **Help.** The following options are available:
  - **Contents and Search.** Displays the Help system.
  - **Version.** Displays information about the release of the server that you are connecting to, and the operating system.
  - **New Features.** Displays new feature documentation.
  - **Release Notes.** Displays release notes documentation.
- **Server Session Log.** Displays the server log.
- **Refresh.** Refreshes the server application directories.
- **Delete.** Deletes the server node from the DMC.
- **Copy Server Node.** Opens the Server Node Configuration dialog box, which displays information for the selected server. The Node Name is prefaced by COPY_ and can be edited to reflect the desired name.
- **Properties.** Provides access to the server connection configuration parameters. For more information, see *Server Node Configuration Dialog Box* on page 153.

Reference: Directory Shortcut Menu Options

Directory shortcut menu options are available if you right-click any of the application directories in the navigation pane. The following options are available:

- **Refresh.** Refreshes the list of objects in the directory.
- **New.** Opens a sub-menu that allows you to create a new object in the selected directory.
- **Paste.** Moves or copies a directory to the selected location. This option is only available after you have copied or cut a directory.
- **Quick Copy.** Creates a simple Data Flow that selects all rows and columns from a source table.
- **Schedule and E-mail.** Has the following options:
  - **Scheduler Agents.** Shows all flows and procedures that contain a schedule.

    Note: This report runs for the user whose ID value is stored in the sched_scan_id of a Scheduler. The Scheduler must be active to run the Scheduler Agents report.

    For more information on the commands available from the Report tab while viewing a Scheduler Agents report, see *Reports Tab* on page 140.

- **Scheduled Events.** Opens a dialog box where you can view jobs scheduled to start during a specified time range. The Scheduler must be active to run the Scheduler Events report.

  When you click View Report, all jobs scheduled to run during the time range are shown.

  For more information about scheduler options, see *Schedule Attribute Pane* on page 410. For more information on email notifications, see *How to Set the Flow Email Notification Properties* on page 147.

- **Log and Statistics.** Opens the Log and Statistics dialog box that enables you to view a log or statistics report. The following options are available:

  - **Report Type.** Allows you to select either a log or statistics report.
  - **Report Range.** Allows you to select to view the last report, all reports, or a report specified within a select time range.
  - **Completion Status.** Allows you to view all reports, all completed reports, all completed successfully reports, or all completed reports that contained errors.
  - **Run by.** Allows you to view reports for all flows, including those initiated by the scheduler or submitted manually. This includes flows initiated from the DMC, WC, and CMRUN.
  - **Application or Flow Name.** Allows you to select a specific application or flow name for the report.

- **Impact Analysis.** Opens a sub-menu that allows you to view impact analysis reports. For more information, see *Impact Analysis Reports on Synonyms, Procedures and Columns* on page 459.
- **Copy.** Copies the selected directory.
- **Delete.** Deletes the selected application directory.
- **Rename.** Allows you to rename the selected application directory.
- **Cut.** Cuts the selected directory.
- **Explorer.** Opens an explorer window in the workspace area.
- **Privileges.** Shows the user privileges associated with the selected directory.

  Note: This option is only available when connecting to a server with security enabled.

- **Properties.** Opens a properties window for the selected directory that displays the following information:
  - **Location.** Physical location of the application directory on the disk.
  - **Number of files.** Number of files in the application directory.
  - **Last modified.** Date and time the last changes were made to the directory.
  - **Description.** A user entered description for the application directory. This is the only field that can be edited.

Reference: Procedures Shortcut Menu Options

On the Home tab, in the Filter group, clicking *Procedures* shows saved flows, procedures, and user functions. You can add flows and procedures to a process flow by dragging them from the navigation pane into the workspace.

If you right-click a flow or a procedure, a shortcut menu opens.

The following options are available:

Open

Adds the flow or procedure to the workspace. A flow opens in the flow editor, while a procedure opens in the text editor.

Edit as Text

Opens a text view of the flow in the workspace. This option is only available for flows.

Note: If you change the text behind the flow, those manual changes will be lost if you edit and re-save the flow in the DMC.

Submit

Submits the flow to the server. This option is only available for a flow. Log information is written to the Data Migrator Log.

Run

Runs the flow or stored procedure immediately. Log information appears in the console log window.

Run Advanced

Has the following options:

Submit with E-Mail

Submits the flow or stored procedure to run in the background. The Submit flow dialog box opens.

The following options are available:

- **Run at a specified date and time.** Allows you to schedule a procedure to run at a specified date and time.
- **Do not wait for completion.** Allows you to continue your work as you wait for the submitted flow or stored procedure to complete. When this option is selected, an additional check box appears which allows you to specify that the flow or procedure should be run at a specified date and time.
- **Submit changing defaults.** Runs a procedure with a prompt to override default values for variables in a procedure or flow, then run it.
- **Submit with Limits.** This option is only available for Data Migrator flows, not stored procedures. Allows you to specify the start and end of the loading records.
- **Report Name.** The optional report name.
- **Schedule Status.** Determines if there is a schedule and if so, whether the schedule is active or inactive.
- **ETLLOG.** Sends the log to ETLLOG and the output to disk.
- **EDAPRINT.** Sends the log to EDAPRINT.
- **E-Mail.** Sends the log and output as an email attachment, if the email notification settings *On Completion* or *On Failure* are requested.
- **Send HTML output as inline message.** When selected, if any stored procedures called by the flow generate an output, the message is included as message text, instead of as an attachment.

  Note: This option for a flow schedule works in conjunction with the *Merge formatted output into log* option that is available for a Scheduler Special Service. The following considerations should be taken into account:

  - When the *Send HTML output as inline message* and *Merge formatted output into log* options are selected, the submitted flow emails the attached ETLLOG file and the HTML output is embedded into the body of the email.

- When the *Send HTML output as inline message* and *Merge formatted output into log* options are not selected, the submitted flow emails the attached ETLLOG file and the HTML output is embedded into the attached ETLLOG file

- **E-mail Notification Condition.** Allows you to specify conditions for sending email. The options are:

  E-Mail On Start. Sends an email when the procedure begins.

  E-Mail On Completion. Sends an email when the procedure completes.

  E-Mail On Failure. Sends an email only if the procedure fails.

  Mail To. Is the field where you enter the email address that the message is being sent to.

  On Completion or Failure, the email message will contain attachments of the log and output from the procedure, if there are any.

  Run Stress. Generates a script for scalability testing. For more information, see the *TIBCO WebFOCUS® Reporting Server Administration* manual.

Schedule and E-mail

Has the following options:

- **Manage.** Allows the schedule and email notification settings for a flow or stored procedure to be changed without opening the flow.

  For more information about Scheduler options, see *Schedule Attribute Pane* on page 410. For more information on email notifications, see *How to Set the Flow Email Notification Properties* on page 147.

- **Scheduler Agents.** This report runs for the user whose ID value is stored in the sched_scan_id of a Scheduler. The Scheduler must be active to run the Scheduler Agents report.

- **Scheduled Events.** Opens a window that shows all jobs scheduled to run during a specified time range. The Scheduler must be active to run the Scheduler Events report.

Logs

Has the following options:

- **Last Log.** Shows the last log available for the selected flow or stored procedure.

- **Last Output.** Shows the last output, if any, for the selected flow.

- **Log and Statistics.** Opens the Log and Statistics dialog box that enables you to view a log or statistics report. The following options are available.

Report Type

Allows you to select either a log or statistics report.

Report Range

Allows you to select to view the last report, all the reports, or a report specified within a select time range.

Completion Status

Allows you to view the log for all flows, all completed flows, all flows completed successfully, or all completed flows that contained errors.

Run by

Allows you to view reports for all flows, including those initiated by the scheduler or submitted by the user. This includes flows initiated from the DMC, WC, and CMRUN.

Application or Flow Name

Allows you to specify a specific application, flow, or procedure to run the report on.

Analysis Reports

Has the following options:

- **Impact Analysis.** Opens a report that shows every flow or stored procedure that calls the selected flow.
- **Dependencies Analysis.** Opens a report that shows all synonyms used by the selected data flow and all flows and procedures used by the selected process flow.
- **Data Lineage.** Opens a report that shows the name of each field used in the procedure, how it is used, its synonym name, and the name of the table and column in the underlying database.

Flow Report

Provides a summary of the attributes of the flow. This option is only available for flows.

Download

Allows you to download the selected file to the Downloads directory in your system.

Cut, Copy, Delete, Rename

Cuts, copies, deletes, or renames the flow or stored procedure.

Privileges

This option is only available if the server is started with security enabled. For more information, see *Manage Privileges Dialog Box* on page 168.

Source Control

This option only appears if you have a source management program configured on your machine. It has the following options:

Add/Check In

Adds or checks files into source management.

Check Out

Checks files out of source management.

Undo Check Out

Cancels a check out.

History

Provides the source management history of a file.

Properties

Opens a Properties dialog box of the selected flow or stored procedure.

Reference: Synonyms Shortcut Menu Options

On the Home tab, in the Filter group, click *Synonyms* to show synonyms that describe the data. You can use them as data sources or data targets in your flow. The synonym type is listed after its name. The type of synonyms available will depend on the adapters that are configured on your server.

If you right-click a synonym, the following options are made available:

Open

Opens the synonym in the workspace. The file, segments, and columns appear on the left, and their attributes and values appear on the right.

Sample Data

Shows a sample of the data associated with the synonym.

Data Profiling

Provides the data characteristics of the selected synonym columns. The following options are available:

Statistics

Provides a data profile of each column of the selected synonym. For more information, see *How to View a Data Profiling Report* on page 536.

Count

Shows a count of the number of rows in the selected source table. For a multi-segment synonym, it shows the count for each segment.

Key Analysis

Analyzes the data source to find out which columns or combination of columns are unique and could potentially be used as key columns.

Analysis Reports

The following options are available:

Impact Analysis

Allows you to assess the impact of a change to the synonym on this server and any subservers that use it. Lists all flows, stored procedures, or reports using the synonym. For data flows, it will indicate whether the synonym is used as a source or a target. Flows and stored procedures are referred to generically as procedures.

Dependencies Analysis

For a synonym that contains a cluster join or multiple tables, all synonyms directly referenced by the selected synonym will be shown.

Metadata Management

The following options are available:

Create Cluster Synonym

Opens a new synonym in the Synonym Editor with a reference to the selected synonym. References to additional synonyms can be added to create a cluster synonym comprising multiple tables.

Refresh

Refreshes the selected synonym. Updates the synonym from the database with changed field information while preserving its old title, description, usage, DEFINE field, and DBA information.

Edit as Text

Places the synonym code in the workspace where it can be edited.

Edit Access File as Text

Places a relational synonym Access File in the workspace where it can be edited. This option is not available for FOCUS files.

Download Master File

Saves the Master File in you Downloads directory.

Download Access File

Saves the Access File in you Downloads directory.

Prepare Translation Files

Opens a dialog box for selecting an application where translation files will be stored, entering characters to be prefixed to the translation file names, and for selecting the configuration file that contains the list of language codes for which translation files should be prepared. Once the translation files are prepared, you must edit each one to have the attribute names in the specified language.

Data Management

The following options are available:

Recreate DBMS Table

Deletes the data and recreates the underlying DBMS table of the synonym.

Synonyms created in Release 7.7.04 or later include information about database indexes.

Note: Use caution when using this command if the database table was created using the data description language of a particular RDBMS. Some attributes of relational database tables, including the column order and database constraints, are not reflected in the synonym, and therefore, cannot be recreated.

Delete ALL Data

Deletes the data from a table or file described by a synonym. This is useful to clear a synonym that will be used as a data target.

Drop Table

Drops the underlying table from the database or deletes a file from the disk.

Insert Sample Data

Loads the synonym with sample records (for relational and fixed format file synonyms). You will be prompted for the number of sample records to load into the table.

If the synonym is for an adapter that supports Bulk Load, a *Bulk Load* check box is available in the Insert Sample Data dialog. If you select the Bulk Load check box and click *Insert*, the sample data will be inserted using Bulk Load. This reduces load times when a large number of sample rows are inserted.

Note: If you are inserting sample data into a fixed format file, the synonym must contain a DATASET statement indicating where to put the data.

Show/Modify Data

Allows you to view and edit data in a database table.

Quick Copy

Creates a simple data flow that selects all rows and columns from a source table. For more information on this option, see *Creating a Quick Copy* on page 367.

Source Control

This option only appears if you have a source management program configured on your PC. It has the following options:

Add/Check In

Adds or checks files into source management.

Check Out

Checks files out of source management.

Undo Check Out

Cancels a check out.

History

Provides the source management history of a file.

For more information regarding source management programs, see *Source Management* on page 709.

Cut, Copy, Delete, Rename

Cuts, copies, deletes, or renames the synonym.

Privileges

This option is only available if the server is started with security enabled. For more information, see *Manage Privileges Dialog Box* on page 168.

Properties

Opens a Properties dialog box of the selected synonym.

Copying, Cutting, and Deleting Procedures, Synonyms, and Other Files

Procedures, synonyms, and other files on the server can be copied, cut, and deleted from the navigation pane.

Note that multiple objects can be selected when performing these procedures.

- To select multiple adjacent objects, click on the first object you wish to select, and then hold the Shift key while selecting the last object.
- To select multiple non-adjacent objects, click on the first object you wish to select, and then hold the Ctrl key while selecting any additional objects.

Procedure: How to Copy Procedures and Synonyms From the Application Directories Folder

From the Application Directories folder of the navigation pane:

1. Open an application directory.
2. Right-click a procedure or synonym and click *Copy*.
3. Right-click an application directory and click *Paste*.

   If you select the same directory, you will be prompted for a different name.
4. Enter a name in the Name field and click *Save*.

Procedure: How to Cut Procedures, Synonyms, and Other Files From the Application Directories Folder

From the Application Directories folder of the navigation pane:

1. Open an application directory.
2. Right-click a procedure, synonym, or other file and click *Cut*.
3. Right-click an application directory and click *Paste*.

   If you select the same directory, you will be prompted for a different name.
4. Enter a name in the Name field and click *Save*.

Procedure: How to Delete Procedures, Synonyms, or Other Files From the Applications Directories Folder

From the Application Directories folder of the navigation pane:

1. Open an application directory.
2. Right-click a procedure, synonym, or other file and click *Delete*.
3. Click *OK* to confirm the deletion.

Workspace

Flows are built and modified in the Data Migrator workspace. Objects can be dragged into the workspace from the navigation pane and, when in process flow mode, from the process flow toolbar.

For data flows and process flows, the workspace consists of the following tabs:

- **Data Flow tab.** Build data flows to extract and move data from data source objects to data target objects. Arrows connect the objects and support the flow.

- **Process Flow tab.** Build process flows. Arrows connect the objects and support the flow.
- **Text View tab.** Allows you to view the underlying code for a flow.

Other tabs are available when creating different types of data flows:

- **Direct Load Flow tab.** Builds direct load flows. For information, see *Creating a Direct Load Flow* on page 363.
- **DBMS SQL Flow tab.** Builds DBMS SQL flows. For information, see *Creating a DBMS SQL Flow* on page 365.

Note: When building different types of data flows, a Process Flow and Text View tab are also available.

For synonyms, the workspace consists of the following tabs:

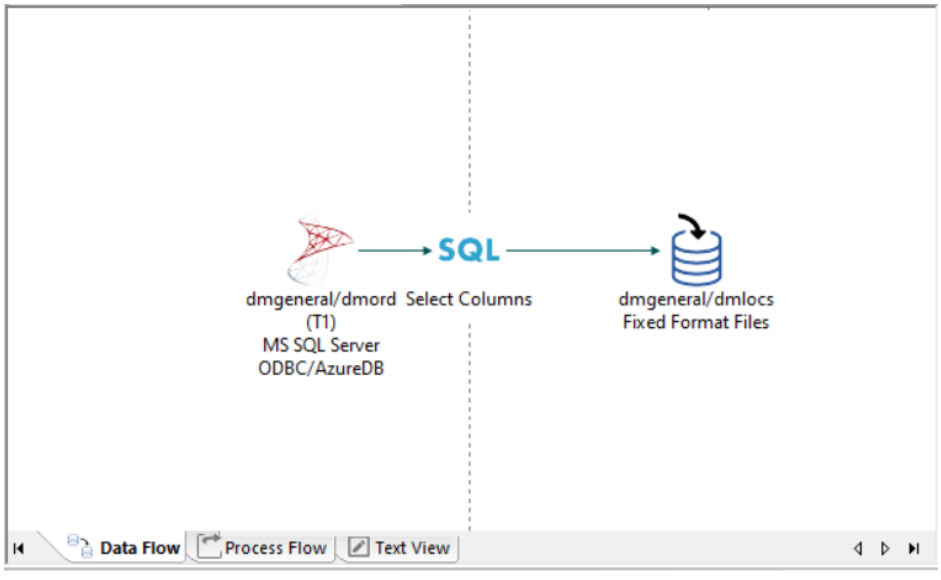
- **Field View.** Shows a combined view of the segment and field names used. A tree of the segments and columns in the synonym are shown on the left, with the attributes and values of the selected item on the right.
- **Segment View.** Shows the name used to identify the specified segment in a tree format.
- **List View.** Shows the field names used in a grid format.
- **Modeling View.** Shows a graphical representation of the synonym, which can be double clicked to access columns, sample data, and sample data with parent keys.
- **Text View.** Shows the code that describes the segments and fields of a synonym.
- **Access File Text View.** Shows the syntax behind an Access File of a synonym.

For details about synonyms, see *Working With Synonyms* on page 485.

Data Flow Tab

The Data Flow tab is where you build data flows to extract and move data from data source objects to data target objects. Arrows connect the objects and support the flow.

The following image is a sample of a simple data flow in the Data Flow tab.



For more information on working with process flows, see *Designing a Data Flow* on page 171.

Procedure: How to Validate a Flow

1. On the Flow tab, in the Tools group, click *Validate*.

   The flow is checked for errors.

2. If there are no errors, the Valid dialog box opens, confirming that no errors or warnings were found.

   If errors are found, a warning message appears.

3. To exit the window, click *OK*. For more information about the error, click *Details* in the Warning window.

Reference: Moving Objects in the Workspace

Multiple objects can be selected in the workspace by selecting them while pressing the Shift or Ctrl keys. You can also hold down the Ctrl key while clicking the right mouse button. This will allow you to move or delete the selected items as a group.

Working With Columns, Variables, and Functions

Data Migrator features a number of calculators for creating transformations and filters using columns, variables, and functions. These columns, variables, and functions can be used in the Synonym Editor to create virtual columns.

These items can be displayed either in a tree or in a list. When displayed in a list, they can be sorted either ascending or descending. You can also search for items.

You can customize the information that is available for columns, as well as how they are displayed.

Note: Tree view is the default for the calculators and for Synonym Editor. List view is the default for Join Properties and Column Selection (SQL object).

Reference: Column Symbols

Next to each column name, there is a symbol that indicates the type of column it is. The columns symbols are:

Symbol	Description
	Key column **Note:** The icon will vary based on the format of the column.
	Regular column **Note:** The icon will vary based on the format of the column.
	Index column
	Virtual column The symbol for columns in the synonym is gray, while the background for a source transformation (virtual column in the flow) is white. **Note:** The icon will vary based on the format of the column.
	Source Transformation **Note:** This is only available in the Source Transformations dialog box. **Note:** The icon will vary based on the format of the column.

Reference: Column Header Shortcut Menu

To access the column header shortcut menu, right-click any column header.

The column header shortcut menu has the following options:

Sort Ascending
Sorts the list of items in the selected column in ascending order.

Sort Descending
Sorts the list of items in the selected column in descending order.

Find

Allows you to search for a column

Field View
Switches views of the synonyms.

List View
Switches views of the synonyms.

Note: When in List View, clicking a column heading will sort the column grid on that heading.

Best Fit

Optimizes the column widths for best viewing.

Customize

Allows you to select what information is displayed for each column.

Column naming display strategy
Controls the information that appears in trees and grids. The available options are Name, Title, and Description. If no title or description exists, the display will default to the alias.

Paste Transforms

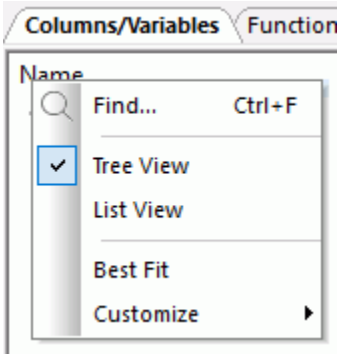
Pastes transformations from memory into the transformation list.

Procedure: How to Change the Display of Columns and Variables in a Calculator

To open a calculator, click the *Insert* icon  or the *Compute* icon  , or right-click the SQL object and click *WHERE Filter* or *HAVING Filter*.

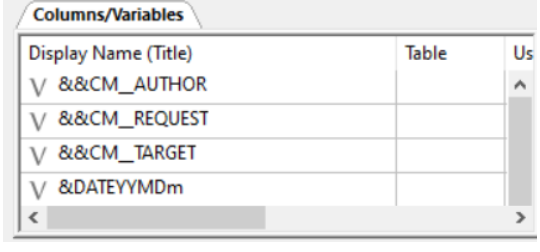
Calculators open to the Columns/Variables tab using the default tree view. To change it:

1. Right-click the *Display Name* heading bar, as shown in the following image.



2. Click *List View*.

   The columns and variables are now shown in a list, as shown in the following image.



Procedure: How to Change the Display of Functions in a Calculator

To open a calculator, click the *Insert* icon or the *Compute* icon, or right-click the SQL object and click *WHERE Filter*.

Calculators open to the Columns/Variables tab using the default tree view.

To change the display of functions:

1. Click the *Functions* tab.
2. Right-click the *Display Name* heading bar.
3. Click *List View*.

   The functions are now shown in a list.
4. Optionally, you can choose *Sort Ascending* or *Sort Descending*.
5. Optionally, you can choose *Best Fit*, which will resize the column to fit its data.

Procedure: How to Change the Display of Columns in the Synonym Editor

Double-clicking a synonym opens the Synonym Editor.

1. Right-click the *Display Name* heading bar.
2. Select *List View*.

   The columns are now shown in a list.
3. Optionally, you can choose *Best Fit*, which will resize the column to fit its data.

Procedure: How to Change the Information Displayed for Columns

The information displayed for columns is controlled by the Column Management pane of the Options dialog box.

In a calculator or the Synonym Editor:

1. Right-click the *Display Name* heading bar.
2. Point to *Customize* and then click a column to add or remove from the shortcut menu.
3. Optionally, click *More* to add or remove multiple columns or reorder them.

   The Column Management pane of the Options dialog box opens.

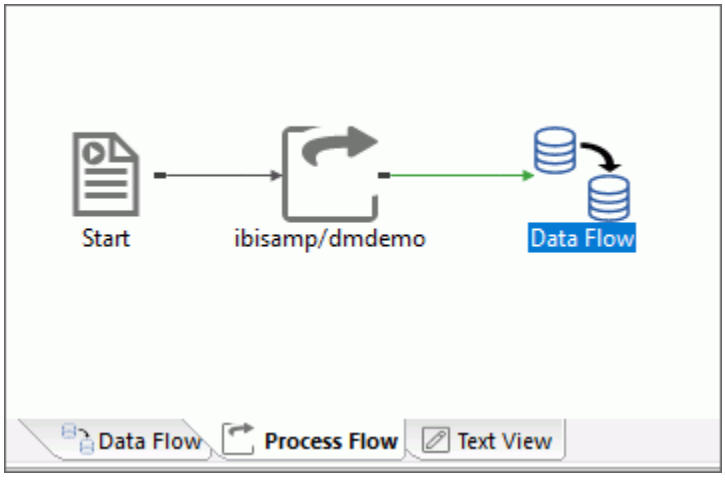
   For more information on the options available in the Column Management pane, see *User Preferences - Column Management* on page 159.
4. Select the columns you want to appear by selecting their check boxes.
5. Optionally, you can change the display order of the columns by moving them up or down in the *Selected Columns* field.
6. Optionally, you can change the *Column Name Display Strategy* to control what appears in trees and grids. By default, the Title is displayed, but you can change it back to Name, which is more readable in calculations. You can have columns listed by Name, Alias, Title, or Description. If no Title or Description exists, the display will default to the Alias.
7. Click *OK*.

   The columns will now appear in the order you specify in calculators or the Synonym Editor.

Process Flow Tab

The Process Flow tab is where you build process flows. Arrows connect the objects and support the flow.

The following image is a sample of a simple process flow in the Process Flow tab.

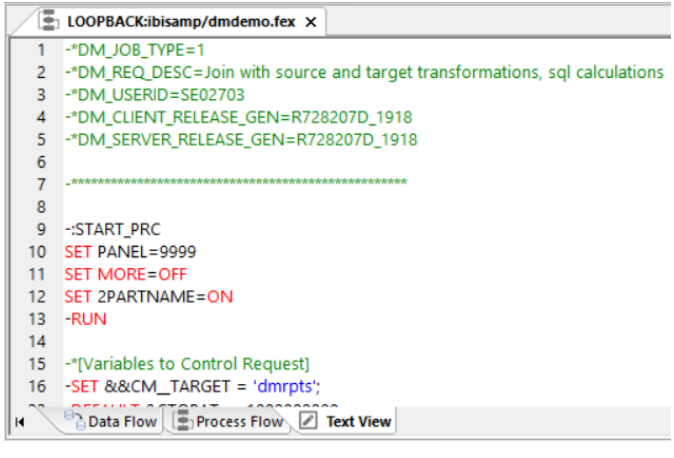


For more information on working with process flows, see *Designing a Process Flow* on page 369.

Text View Tab

The Text View tab allows you to view the underlying code for a flow.

Note: You can use Ctrl + F or select *Find* from the Edit menu to search the text. After you close the Find dialog box, you can press the F3 key to find the next occurrence and Shift + F3 to search backwards.



Text Editor

The Data Management Console has a graphical user interface for editing flows, synonyms, and workspace configuration. It is recommended that this interface should be used since it ensures proper syntax. However, for advanced users, the Data Management Console also has a text editor that can be used to edit the underlying files.

Note: Changes made to a flow using the text editor may be lost when the flow is opened normally.

Use caution when adding comments to the first lines (header) of a stored procedure or a flow. Avoid comments starting with -*DM, since such comments are reserved for Data Migrator internal use. A best practice is to put a space between -* and the rest of your comment.

Reference: Text Editor Shortcut Menu

Depending on where you right-click in the Text Editor, the following shortcut menu options are available:

Undo

Undoes the last edit operation.

Redo

Undoes the last undo.

Cut

Cuts the selected text to the clipboard.

Copy

Copies the selected text to the clipboard.

Delete

Deletes the selected text.

Paste

Inserts text from the clipboard in the specified location.

Select All

Selects the entire file.

Find

Opens the Find dialog box, which allows you to enter a text string and search parameters.

Find Next

Finds the next occurrence.

Find Previous

Finds the previous occurrence.

Replace

Opens the Replace dialog box, which allows you to enter a find and replace string.

Go To

Opens the Go To dialog box, which allows you to enter a specific line number to go to.

Change Case

This option is available when text is selected. The following options are available:

- **Invert Case.** Changes all uppercase letters to lowercase and all lowercase to uppercase. For example, sERVER will be changed to Server.
- **Upper Case.** Changes selected text to all uppercase.
- **Lower Case.** Changes selected text to lowercase.

Comment Selection

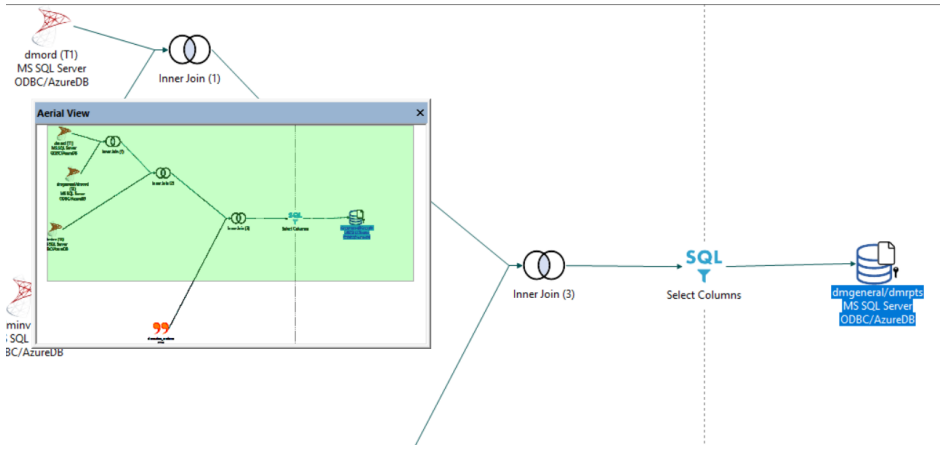
Adds the comment symbol -* in front of the selected lines. This option is only available for procedures.

Uncomment Selection

Removes the comment symbol -* when it appears at the start of the selected lines. This option is only available for procedures.

Aerial View

Aerial View is available in the View main menu option by clicking *Aerial View*. For large and complex process flows, data flows, and Synonym Editor Modeling Views, the Aerial View allows you to select the portion of the flow or synonym that appears in the workspace. The green area in the Aerial View shows the area that is viewed, as shown in the following image.



Ribbon Tabs

The ribbon represents the activities performed in the DMC. Commands are active or inactive depending on the activity being performed. These include a general toolbar for operations like opening and saving files, a process and data flow toolbar for operations like running and deleting flows, a specialized process flow toolbar for adding objects to the workspace, and a reports toolbar for operations like displaying data in a spreadsheet.

Home Tab

The Home tab consists of the following groups and commands:

View Group

Status Bar

Displays the status bar at the bottom of the DMC.

Browser

Displays the browser in the DMC.

Console Log

Displays the Console Log in the DMC.

Full Screen

Displays the workspace in full screen mode.

Explorer

Opens a Windows Explorer style window on the right side of the DMC that shows detailed information about the currently selected server object.

Back

Goes back one directory level. This option is available when in Explorer view.

Forward

Goes forward one directory level. This option is available when in Explorer view.

Up

Goes up one directory level. This option is available when in Explorer view.

Filter Group

Clear

Clears any of the filters set for an application directory.

Procedures

Shows saved flows, procedures, and user functions in the application directory.

Synonyms

Shows synonyms that describe data.

Advanced

Allows you to specify what you want to see in the navigation pane. For more information, see *Advanced Filter* on page 173.

Find File Group

Find What

Allows you to specify a term to search for in an application directory. You can specify that the search needs to match a whole word or case, by selecting the appropriate check box below the find field.

Tools Group

Console Log File

Opens the console log file in a text editor, such as Notepad.

Selecting the arrow opens a shortcut menu with the following additional options:

- **View Console Log File.** Opens the console log file in a text editor, such as Notepad.
- **Clear Console Log Window.** Deletes all information from the message window.
- **Clear Console Log File.** Deletes all information from the message window log file.

Language Selection

Opens the Language Selection window where you can configure a localized version (for selected languages) of the Data Management Console and the language used for text entry and display.

Options

Opens the Options dialog box where you can set user preferences. For more information, see *Setting User Preferences* on page 156.

Configure Source Control

Opens the Configure Source Control dialog box where you can select a SCC Provider if you have one installed on your PC.

Windows Group

Windows

Has the following options from the shortcut menu:

- **Close.** Closes the active window in the workspace.
- **Close All.** Closes all the open windows in the workspace.
- **Close All But This.** Closes all open windows in the workspace except for the one currently open.

MDI Tabs

Has the following options from the shortcut menu:

- **New Horizontal Tab Group.** Creates a new horizontal tab group and opens the last object there.
- **New Vertical Tab Group.** Creates a new vertical tab group and opens the last object there.

- **Move to Next/Previous Tab Group.** Changes the focus to the next or previous tab group. This option is available only if multiple tab groups are open.

Flow Tab

The Flow tab consists of the following groups that apply to both a data and process flow. Some of these commands are only available when working with a process flow.

Undo/Redo Group

Undo

Undoes the last operation.

Redo

Undoes the last undo.

Clipboard Group

Delete

Deletes the selected object.

Cut

Cuts the selected item.

Copy

Copies the selected item.

Paste

Pastes the cut or copied item.

Editing Group

Find

Opens the Find dialog box, which allows you to enter a text string and search parameters. This option is available when in Text or Edit as Text view.

Find Next

Finds the next occurrence. This option is available when in Text or Edit as Text view

Find Previous

Finds the previous occurrence. This option is available when in Text or Edit as Text view.

Select All

Selects all of the text in the file. This option is available when in Text or Edit as Text view.

Replace

Opens the Replace dialog box that lets you enter a text string to find and replace it with. This option is only available when in Edit as Text view.

Go To

Opens the Go To Line dialog box, which allows you to enter a specific line number to view. This option is available when in Text or Edit as Text view.

Invert Case

Changes all uppercase letters to lowercase and all lowercase to uppercase. For example, sERVER will be changed to Server.

Upper Case

Changes selected text to all uppercase.

Lower Case

Changes selected text to all lowercase.

Comment Selection

Adds the comment symbol -* in front of the selected lines. This option is only available for procedures.

Uncomment Selection

Removes the comment symbol -* when it appears at the start of the selected lines. This option is only available for procedures.

View Group

Data Flow

Switches to the Data Flow view in the workspace.

Process Flow

Switches to the Process Flow view in the workspace.

Text View

Switches to the Text view in the workspace.

Aerial View

Opens the Aerial View window.

Find Flow Object Group

Find What

Allows you to specify a term to search for in the workspace. You can specify that the search needs to match a whole word or case, by selecting the appropriate check box below the Find What field.

Insert Group (Data Flow Only)

If you are working with a data flow, the Insert group consists of the following commands.

Source

Creates a source in the workspace.

Join

Joins two objects in the workspace.

Select

Creates a Select object in the workspace.

Union

Creates a Union object in the workspace.

Existing Target

Adds an existing target in the workspace.

New Target

Creates a new target in the workspace.

Insert Group (Process Flow Only)

If you are working with a process flow, the Insert group consists of the following commands.

Procedure

Creates a Stored Procedure object in the workspace.

Flow

Creates a Flow object in the workspace.

E-Mail

Creates an Email object in the workspace.

Parallel Group

Creates a Parallel Group object in the workspace.

Wait

Creates a Wait object in the workspace.

Set Variables

Sets variables for the process flow.

Run Group

Run

Launches a drop-down menu for Submit, Run, Run Advanced, Impact Analysis, Dependencies Analysis, and Data Lineage.

Reports Group

Impact Analysis

Runs the impact analysis report for the flow that is currently open.

View Last Log

Opens the last log for the flow that is currently open.

View Flow Report

Runs the flow report for the flow that is currently open.

Tools Group

Arrange Items

Rearranges the objects in the workspace.

Validate

Validates the flow.

Properties

Displays the property pane for the flow.

User Functions Tab

The User Functions tab becomes active when you edit functions. It contains the following additional groups:

View Group

Tree View

Shows the selected function in a tree view.

List View

Shows the selected function in a list view.

Text View

Shows the selected function in a text view.

Insert Group

Function

Creates a new function for the function.

Parameter

Creates a new parameter for the function.

Define

Creates a new define for the function.

Run

Test Function

Opens the Test Function dialog box, where you can enter values to test the function.

Tools Group

Properties

Opens the Properties pane for whatever is selected in the workspace.

Reports Tab

The Report tab becomes active when you run a report or view the Server Session Log. The Report tab has the following additional groups.

Print Group

Print

Opens the Print dialog box to print the report.

Print Preview

Displays the preview of the item to be printed.

Run Group

Refresh

Reruns the report.

Next Result

Displays the next report in the output.

Note: This option is only available for output with multiple pages.

Automatic Refresh

Automatically reruns the report at the refresh interval specified.

Refresh Interval (in seconds)

Allows you to specify how often a report reruns.

Tools Group

Export to Excel

Exports the report as an Excel spreadsheet.

Note: This option is available for Alpha and Original output only.

Copy as Text

Copies the selected report text to the clipboard.

Filter

Allows you to specify what information is shown in the report. The options available are FOCUS, SQL/Mdx, Transform, Errors, Messages, Timestamp, and No Delay. This option is available when viewing a Server Session Log.

Clear Messages

Deletes all messages from the log and controls what types of messages appear in the server session log. This option is available when viewing a Server Session Log.

Choose Columns

Opens the Scheduler Agent page where you can select the columns that appear on the scheduler agents page. This option is available when viewing the Scheduler Agent.

Choose States

Opens the Select the states that will appear on the Scheduler Agents dialog box where you can select what states will appear on the Scheduler Agents page. This option is available when viewing the Scheduler Agent.

Activate Selected

Allows you to select multiple flows or procedures and set the schedule status as active in a single operation. When using this command, it is recommended that you use Choose columns and select the Status column to display. This option is available when viewing the Scheduler Agent.

Deactivate Selected

Allows you to select multiple flows or procedures and set the schedule status as inactive in a single operation. When using this command, it is recommended that you use Choose columns and select the Status column to display. This option is available when viewing the Scheduler Agent.

Metadata Tab

The Metadata tab becomes active when you are working with a synonym. For more information, see *Working With Synonyms* on page 485.

Text Editor Tab

The Text Editor tab becomes active when you are using the Text Editor. It contains the following additional group:

Run Group

Run

Launches a drop-down menu for Run, Run with Options, and Impact Analysis.

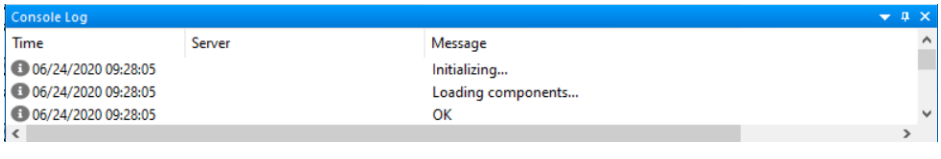
Console Log

The Data Migrator Console Log shows results from the DMC communications with the server.

The Console Log displays:

- Log messages when you connect or disconnect from the server, run reports, or save flows.
- Server messages when you execute flows or stored procedures.

To toggle the Console Log, select or deselect *Console Log* from the View menu or press Alt+9.



If you right-click the console log, the following options are available:

View Console Log File

Displays the log file in Notepad.

Clear Console Log Window

Deletes all information from the message window.

Clear Console Log File

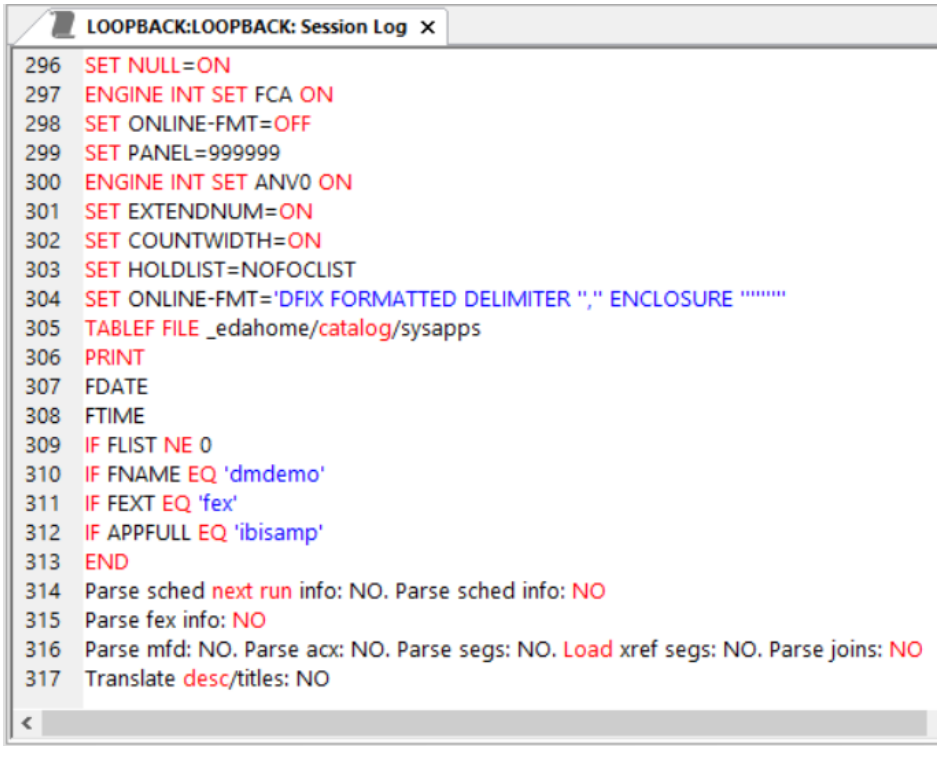
Deletes all information from the message window log file.

Copy Selection

Copies items highlighted in the Message window to the clipboard. Note that you can select multiple lines by pressing the Shift or Ctrl key while selecting each line.

Session Log

The Server Session Log shows the actual commands that are sent from the DMC to the server, which may be useful for problem determination. It is available by right-clicking a server in the navigation pane and clicking *Session Log* from the shortcut menu. The log will then open in the workspace, as shown in the following image.



For more information on the commands available when viewing this report, see *Reports Tab* on page 140.

Setting Flow Properties

Flows properties allow you to view or set properties that apply to the entire data flow. These include:

- **General properties.** Includes who created the flow, the last time it was updated, a description, and comments.
- **Execution properties.** Includes stop processing options.
- **Restart properties.** Includes the number of restart attempts and their start points. A restart attempt occurs if the flow execution fails. Restart attempts can be made from the first record or from the last commit point.
- **Record Logging.** Shows actions to be taken when processing records with errors.

Specify flow properties in the Properties pane by clicking *Properties* in the Tools group of the Flow tab.

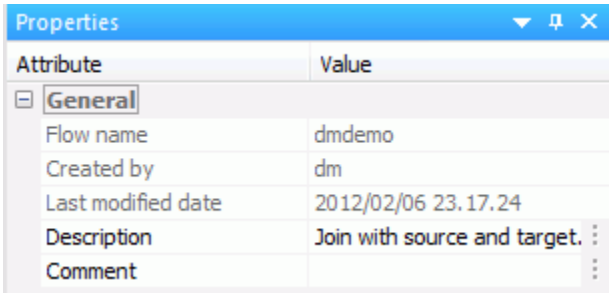
If you right-click anywhere in the Properties pane, a shortcut menu opens, with the following options available:

- **Categorized.** Displays all of the available properties for the selected flow by category.
- **Alphabetic.** Displays all of the available properties for the selected flow in alphabetic order.
- **Find.** Allows you to search for text in the Attributes column.

Procedure: How to Set the Flow General Properties

1. On the Flow tab, in the Tools group, click *Properties*.

   The Properties pane opens, as shown in the following image.



   The General attribute section of the Properties pane has the following fields and options:

   Description

   Is a description of the flow. You can enter up to 150 characters. Descriptions appear in the navigation pane next to the flow name.

   Comment

   Are comments regarding the flow.

   This section also provides information on the flow name, the user who created the flow, and the date it was last modified.

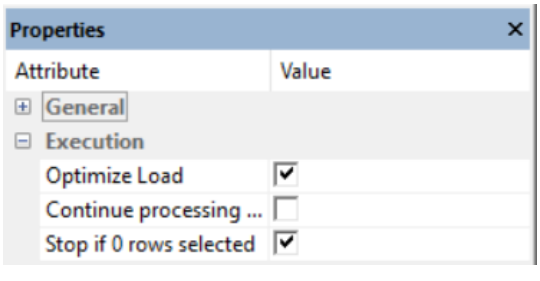
2. Make your desired changes and click *Save* in the Quick Access Toolbar.

Procedure: How to Set the Flow Execution Properties

1. On the Flow tab, in the Tools group, click *Properties*.

   The Properties pane opens.

2. Expand the *Execution* attribute, as shown in the following image.



The Execution attribute section of the Properties pane has the following fields and options:

Optimize Load

Allows faster loading by combining insert and update operations in a flow. It also uses a more efficient algorithm in a Slowly Changing Dimensions flow. This option can be used with databases that use the MERGE command, as well as Insert only operations. When you create a new flow in the DMC, this option is selected, by default. If you do not want to use this option, you must deselect it.

Continue processing when a partial answer set is returned

When selected, a flow will continue to run, even if an error is generated.

Stop after DBMS errors

Causes the flow to fail with a non-zero return code if the number of errors specified is exceeded. Applies to relational targets when Insert/Update or Insert records from memory is used. Any rows written to the RDBMS are committed to the RDBMS. The default value is 10000000, unless this is overridden in the Run Options found in the Tools group of the Home tab.

A relational database may return an error for a constraint violation, such as a Unique Index constraint, a Not Null constraint, or a user-specified constraint.

If such errors are not expected, you may want to fix their cause without waiting for the entire flow to be processed. Using a low value, such as 10, can provide enough information to determine the cause of the problem, without running the entire job.

If such errors are expected, and you want to continue processing anyway, use a high value, one greater than the number of records in the source, or do not select the *Stop processing after* check box.

Note:

- When using the Load Type setting of Bulkload utility using a disk file, this setting is not used.
- This setting is not available if the Optimize Load option is enabled.

Stop if 0 rows selected

Allows you to stop processing if no data is returned. When this option is selected, and no data is returned, code 18708 occurs.

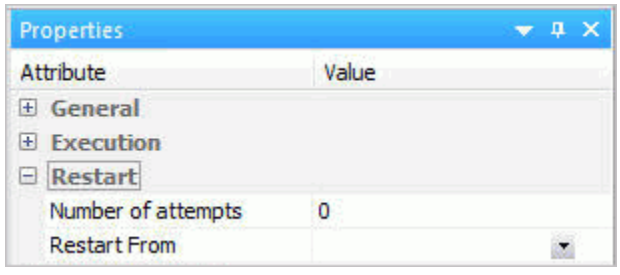
3. Make your desired changes and click *Save* in the Quick Access Toolbar.

Procedure: How to Set the Flow Restart Properties

1. On the Flow tab, in the Tools group, click *Properties*.

   The Properties pane opens.

2. Expand the *Restart* attribute, as shown in the following image.



   The Restart attribute section of the Properties pane has the following fields and options:

   Number of Attempts

   Is the number of times a flow will be restarted if the execution fails.

   Note: The amount of time between restart attempts can be set for the server. The default value is 60 seconds. For more information, see *Setting Up TIBCO Data Migrator* on page 27.

   Restart From

   Is the point from which the next attempt will be made. The options are *Beginning* or *Last Commit*.

   Note: Last Commit cannot be used for fixed format file, delimited file, or XML targets. Last Commit can only be used for existing targets, and cannot be used with Prior to Load Options of *Delete all rows from table*, *Truncate table*, or *Drop table*.

3. Make your desired changes and click *Save* in the Quick Access Toolbar.

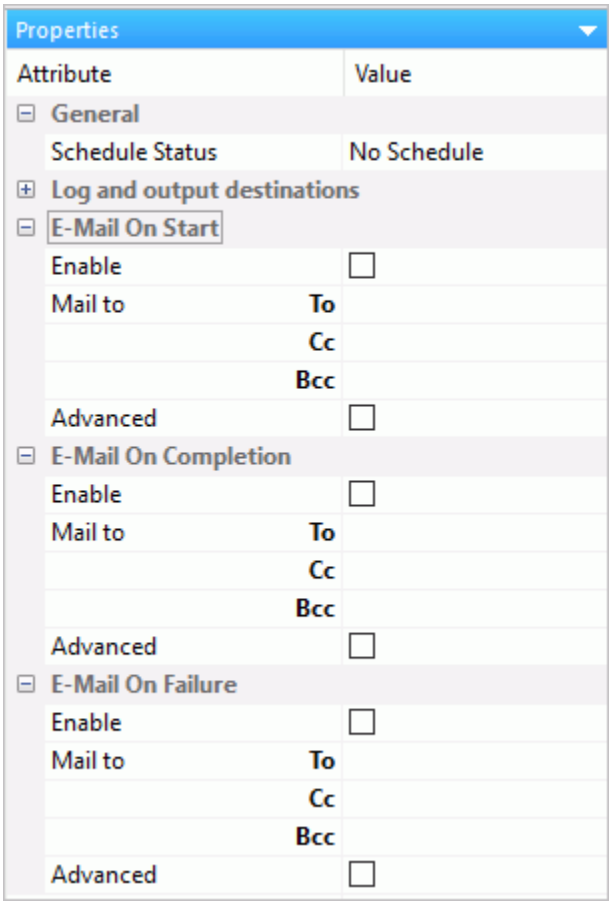
Procedure: How to Set the Flow Email Notification Properties

Note: Your Server must be configured for email notification to use this option.

1. Right-click the Start object in a process flow and click *Properties*.

   The Properties pane opens.

2. The E-mail attributes open, as shown in the following image.



The Log and output destinations section has the *Send HTML output as inline message*. When selected, if any stored procedures called by the flow generate an output, the message is included as message text, instead of as an attachment.

Note: This option for a flow schedule works in conjunction with the *Merge formatted output into log* option that is available for a Scheduler Special Service. The following considerations should be taken into account:

- When the *Send HTML output as inline message* and *Merge formatted output into log* options are selected, the submitted flow emails the attached ETLLOG file and the HTML output is embedded into the body of the email.

- When the *Send HTML output as inline message* and *Merge formatted output into log* options are not selected, the submitted flow emails the attached ETLLOG file and the HTML output is embedded into the attached ETLLOG file

E-Mail On Start, E-Mail On Completion, and E-Mail On Failure all have the following fields and options:

Enable

Select this to enable when an email should be sent.

Mail To

The email addresses to which the messages will be sent. Place each email address (or user ID configured on the server with an associated email address) on a new line with no separators.

Advanced

Select this option to get the following options:

- **Mail to addresses defined in procedure.** You can send the messages to email addresses defined in the procedure by either typing in the location of the procedure, or clicking the ellipses and manually selecting it from the Select Procedure dialog box. For more information, see *How to Send an Email to Addresses Defined in a Procedure* on page 149.
- **Importance.** Select the importance for the email from the drop-down menu. Choices are Low, Normal, or High.
- **Subject.** The subject line of the message.
- **E-Mail Message.** The body of the message.

Note: Your Server must be configured for email notification in order to use this option. See *How to Configure the TIBCO WebFOCUS Reporting Server for Email Notification* on page 71.

3. Make your desired changes and click *Save* in the Quick Access Toolbar.

Procedure: How to Send an Email to Addresses Defined in a Procedure

1. Create a file with the list of the email recipients.

   For example, the following file is called *addresses.txt*.

```
addresses
user_one@tibco.com
user_two@google.com
user_three@eserve1.com
```

This list can be considered a delimited file, even though it has only one column and no delimiters.

2. Create a synonym for the file containing the email addresses, specifying that it has a header line.
   For example, call the synonym *addresses*.

3. Create and save a new stored procedure to read the email addresses.

   For example, the following procedure is called *getaddreses*.

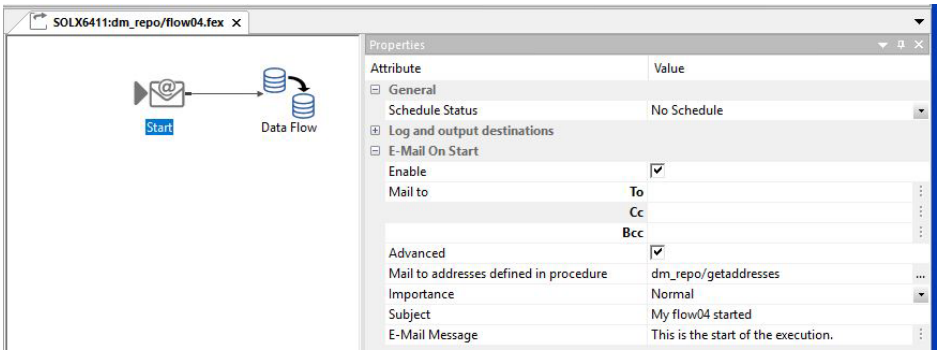
```
TABLE FILE ADDRESSES
PRINT ADDRESS
ON TABLE PCHOLD FORMAT COMT FORMATTED
END
```

4. Open a flow and switch to the Process Flow tab.

   a. Double-click the *Start* object to open its properties.

   b. In *E-Mail on Start*, *E-Mail on Completion*, or *E-Mail on Failure*, select the check boxes Enable and Advanced.

   c. In the *Advanced* section, fill in the value for *Mail to addresses defined in procedure* by entering the application directory and procedure name you just created (for example, baseapp/getaddresses) or by clicking the ellipses and manually selecting the procedure from the *Select Procedure* dialog box, as shown in the following image.



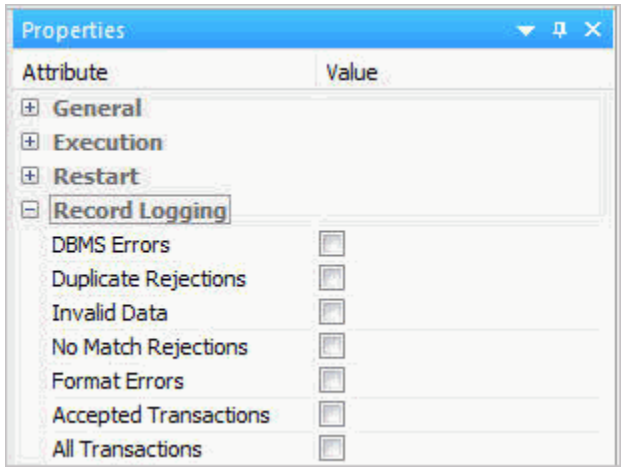
Procedure: How to Set the Flow Record Logging Properties

Note: These settings are not available if the Optimize Load option is enabled.

1. On the Flow tab, in the Tools group, click *Properties*.

   The Properties pane opens.

2. Expand the *Record Logging* attribute, as shown in the following image.



The Record Logging attribute section of the Properties pane has the following fields and options:

DBMS Errors

Logs transactions rejected by an RDBMS due to a constraint violation or data type error. For fixed format file targets, format errors are also recorded.

Duplicate Rejections

Logs transactions rejected due to a duplicate key.

Invalid Data

Logs transactions rejected for not meeting a VALIDATE condition

No Match Rejections

Logs transactions rejected when a required key is not found.

Format Errors

Logs transactions rejected due to a format error. Format errors only apply to direct load flows.

Accepted Transactions

Logs transactions that were accepted.

All Transactions

Logs all transactions.

3. Make your desired changes and click *Save* in the Quick Access menu.

Viewing Properties

During the course of your DM design work, you may need to view and edit the properties of your server connection. For example, you may wish to:

- Update expired user IDs and passwords.
- Change port numbers.

You can perform these server-connection related tasks from the Server Node Configuration dialog box within the DMC.

In addition, you can view, but not edit, the properties for the Application Directories folder, specific application directories within this folder, and procedures within an application directory.

Procedure: How to View Server Information

Server Information displays information about the release of the server to which you are connecting. It is used primarily for customer support, if you ever need to report a problem.

From the navigation pane, right-click the Server, point to *Help*, and then click *Version*.

Procedure: How to View the Server Properties

The Server Node Configuration dialog box shows the parameters you set during installation or when configuring an additional server node. See *How to Install the Data Management Console* on page 32 and *How to Add a Node to Identify the TIBCO WebFOCUS Reporting Server to the Data Management Console* on page 35.

From the navigation pane:

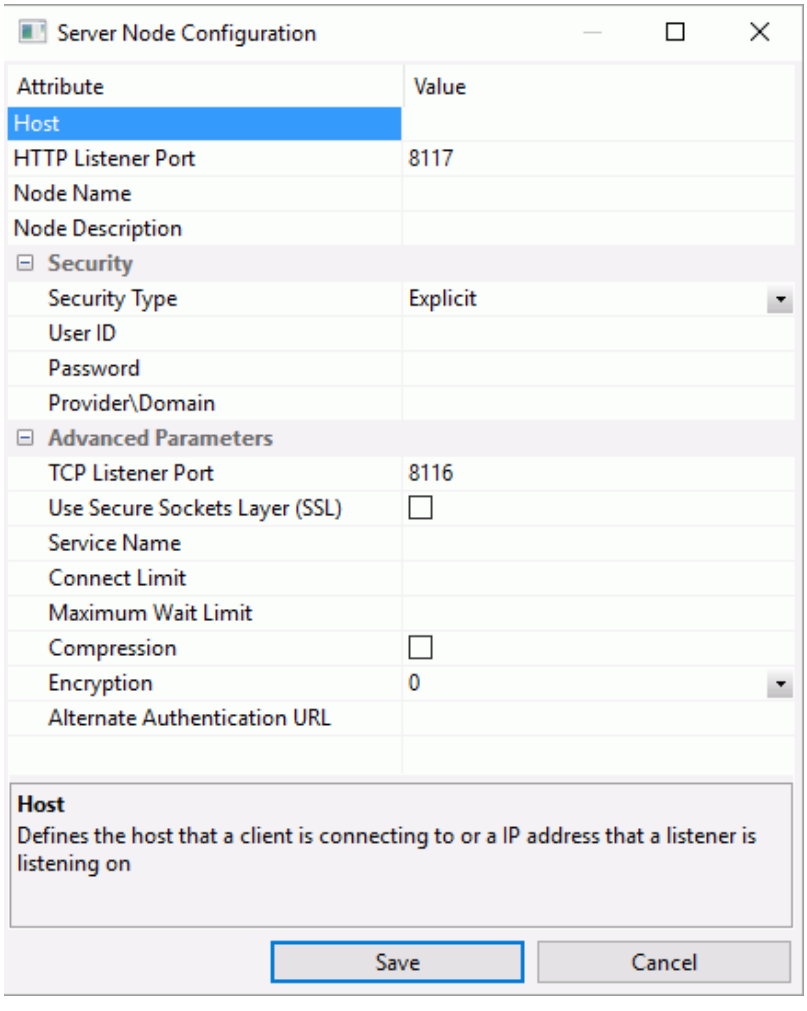
1. Right-click the server and click *Properties*.

   The Server Node Configuration dialog box opens.

2. Make any desired changes.
3. Optionally, if you are using the EDA API to connect an application like TIBCO iWay® Service Manager to the server, expand the *Advanced Parameters* section to set connection and wait limits.
4. Click *Save*.

Reference: **Server Node Configuration Dialog Box**

To access the Server Node Configuration dialog box, right-click the server and click *Properties*.



Note: The general and security settings in the Server Node Configuration dialog box are used by the DMC.

The Server Node Configuration dialog box contains the following fields and options:

Host

Host

Is the machine name or IP address where the Server is running.

HTTP Listener Port

The DMC connects to the server using the HTTP Listener Port. When you install a Server, the HTTP Listener is assigned to port 8121 by default. When you add new server nodes using the DMC, the default port is 8101.

Node Name

Is the Server name.

Node Description

Provides additional information about the node that will be visible in the navigation pane.

Security

Security Type

Is the security setting.

The following options are available:

- **Explicit.** With this setting, if security is required, you must enter it either at connection time or in the Security credentials fields to the right. This is the default value.
- **NTLM.** Use when connecting to a Windows server. The Windows user ID and password to the server are passed to the server when a connection is made.

  Note: Only when OPSYS security is used.
- **Kerberos.** Use when connecting to a Linux server. The user ID and password to the server are passed to the server when a connection is made.

  Note: Only when OPSYS security is used.

Note: Settings on the Server Node Configuration Advanced section are used by applications connecting to the server using the EDA API, including Resource Analyzer and Resource Governor.

Advanced Parameters

TCP Listener Port

Allows you to set your own TCP port, rather than the default, when connecting an application to the server using the EDA API. Applications like Resource Analyzer and Resource Governor use the EDA API to connect to the server through this port. When you install a Server, the TCP Listener is assigned to port 8121 by default. When you add new server nodes, the TCP Listener Port defaults to one less than the HTTP Listener Port.

Use Secure Sockets Layer (SSL)

Sets a Secure Sockets Layer (SSL), which is a commonly-used protocol for managing the security of a message transmission over HTTP. To use the HTTP Listener Secure Sockets Layer, you must first configure the HTTP Listener for SSL.

Service Name

Is the name of the service to which the user ID is connecting. If it is left blank, the user ID will connect to the default service.

Connect Limit

The maximum time that the client will wait for a TCP connection response from the server.

Maximum Wait Limit

The time that the client will wait for a response from the server in seconds.

Compression

Activates data compression in data transfer between the client and the server.

Encryption

Defines the encryption method used in data transfer between the client and the server. Encryption is not supported in PIPE protocol. Only DES is supported in HTTP protocol.

Application Directory Properties

View the properties of an application directory using any of the following methods:

- In the navigation pane, right-click the *Name* heading bar, point to *Customize*, and then click *More*.

The Column Management window opens. This is where you can customize what property columns are displayed in the navigation pane.

- In the navigation pane, right-click *Application Directories* and click *Explorer*.

  The application directories that are configured are now displayed in the workspace. Right-click the heading bar in the workspace, point to *Customize*, and then click *More*.

  The Column Management window opens. This is where you can customize what property columns are displayed in the navigation pane.

For more information on the Column Management window, see *User Preferences - Column Management* on page 159.

Flow or Procedure Properties

View the properties of a flow or procedure using any of the following methods:

- In the navigation pane, right-click the *Name* heading bar, point to *Customize*, and then click *More*.

  The Column Management window opens. This is where you can customize what property columns are displayed in the navigation pane.

  Expand the application directory to see the properties for the contents of that directory.

- In the navigation pane, right-click an application directory and click *Explorer*.

  The contents of the directory are now displayed in the workspace. Right-click the heading bar in the workspace, point to *Customize*, and then click *More*.

  The Column Management window opens. This is where you can customize what property columns are displayed in the navigation pane.

For more information on the Column Management window, see *User Preferences - Column Management* on page 159.

Setting User Preferences

User preferences allow you to set up your DMC environment. On the Home tab, in the Tools group, click *Options* to access these settings.

To access the Options dialog box, on the Home tab, in the Tools group, click *Options*.

The Options dialog box opens with General settings in the right pane. The available options for each area are detailed in the following sections.

Reference: User Preferences - General

General user preferences include using background images, automatically switching and synchronizing trees, determining the location of messages, and controlling scrolling speed.

On the Home tab, in the Tools group, click *Options*, and then click *General* to access these settings.

The General settings dialog box has two sections: General and Others.

The General section has the following fields and options:

Use background images

Set a blue background image for the workspace.

Automatically switch filters

Switches filters automatically depending on what flow view is open. Clicking the Data Flow tab will switch on the Synonyms filter, while clicking the Process Flow tab will switch on the Procedures filter.

Start LOOPBACK server on startup

This option is enabled if you start the DMC from a Windows installation. The local LOOPBACK server will start automatically, if it is not already running, when you start the DMC. When you close the DMC, the LOOPBACK server is stopped. This option is not enabled if you start the DMC from a client installation.

Show tabbed view selector at top of each frame

Moves workspace tabs, like Data Flow, Process Flow, and Text View, to the top. If this option is not selected, by default, the workspace tabs appear at the bottom of the frame.

Auto Save for existing files (in minutes)

You can select the number of elapsed minutes before open files are automatically saved. By default this feature is not enabled.

The Others section has the following options:

Scrolling speed

Determines the speed at which a flow scrolls when you drag an object outside the viewable window. It can be set in relative units of 1 to 10. The default is 1.

Recently used file list

You can select the number of files to display in the recently used file list from 0 to 16. The default is 4.

Display WebFOCUS Environments*

If not selected, the WebFOCUS Environments option is disabled and does not appear in the navigation pane. This option requires restarting the DMC.

Include images when exporting data to Excel

Enables data to be exported with images included.

Clear User Preferences*

Clears all user set preferences and returns them to default values. This option requires restarting the DMC.

Save GUI State

Used for diagnostics, this option saves the current state. This file is only needed if requested by customer support.

Ribbon*

Select this option to enable the Ribbon. Deselect it to use the classic toolbar. This option requires restarting the DMC.

Reference: User Preferences - Format

Format user preferences include settings, colors, and fonts for the Data Console interface.

On the Home tab, in the Tools group, click *Options*, and then click *Format* to access these settings.

The Fonts Category section has the following fields and options:

Category list box

Allows you to set the font for text in the process flow workspace, reports, text views, and log views.

Font

Launches a dialog box for changing font settings.

Reset fonts to defaults

Restores the default fonts.

Display full component names

Allows you to display the full name of a component.

Reference: User Preferences - Column Management

Column Management user preferences enable you to select which columns to display on the grids for transformations, column selection, joins, sorts, sources, and targets. You can also set the column display order. The settings apply to all column-related dialog boxes.

On the Home tab, in the Tools group, click *Options*, and then click *Column Management* to access these settings.

The Column Management settings include the following fields and options:

Customize column display

Lists columns that can be added to the grid using the check boxes.

Column Name Display Strategy

Controls the information that appears in trees and grids. The available options are Name, Title, and Description. By default, the Title is displayed, but you can change it back to Name, which is more readable for calculations. If no Title or Description exists, the display will default to the Alias. When the strategy is something other than the Name, this is indicated in the header row. For example, Display Name (Title).

Function display strategy

Controls how functions are identified on the functions tab in the calculators. By default, the function syntax is displayed. It can be changed to a short description, which displays a brief description of what the function does instead.

Use segment to qualify field reference

Enables you to fully qualify field names in the JOIN criteria of a cluster synonym. The available options are Always and For duplicate fields. The default value is For duplicate fields, which allows you to maintain compatibility with earlier releases.

The default settings are shown. Not all columns are on every grid. The following are examples of columns that can be added:

Expression

Is the expression for the column.

Type

Is the type of object in an application directory.

Length

Is the column length.

Scale

Is the maximum number of digits to the right of the decimal.

Alias

Assigns an alternative name for a column, or the real column name for a DBMS synonym.

Title

Supplies a title to replace the column name that is normally used in reports.

SCD Type

Is used for processing slowly changing dimensions.

Field Type

Indicates that a field is an index (I) and/or is read-only.

Table

Is the synonym that contains the column.

Format

Is the type and length of column data as stored.

Description

Is a description or comment about the column.

Nulls

Indicates whether the column can contain null data.

Note: You can change the display order of the columns by moving them up or down using the arrows. You can also change the *Column Name Display Strategy* to control what appears in trees and grids.

Reference: User Preferences - Traces

On the Home tab, in the Tools group, click *Options*, and then click *Traces* to access these settings.

The Traces settings dialog box has the following options:

Enable

Select this option to perform tracing of the application to debug problems. You can choose the components activated in the trace files tscom3.trc and connection.trc. These trace files reflect the operations between the client and the server. You can view these trace files from the DMC.

View Trace Files

Select which file to view from the drop-down menu. The options are tscom3.trc and connection.trc.

View

Click this button to view the trace file.

Choose the trace components to activate

Allows you to specify which components will be traced.

Reference: User Preferences - Data Flow Designer

Data Flow Designer user preferences include using application directories, automatically selecting and mapping all columns, adding Join objects and conditions, and performing validation.

On the Home tab, in the Tools group, click *Options*, and expand *Data Flow*. Then, click *Data Flow Designer* to access these settings.

The Data Flow Designer settings dialog box has two sections: On Add Source and On Add Target.

Use application directory name with flow components

Includes the application directory name for objects in data flows.

Show Notes

Allows you to add comments to data flow objects, directly from the workspace.

Undo/Redo Limit

Specifies the maximum number of undo operations allowed in the Data Flow Designer.

The On Add Source section has the following fields and options:

Automatically select all columns

Selects all columns in a source.

Add Join Object if needed

When two sources are dragged onto the workspace, this option connects the sources to each other and to the SQL Object using Join objects. This is the default.

Automatically add join conditions

Creates an inner join if two data sources have a column with a primary/foreign key relationship or with the same name and format. The column must be a key column in one of the sources. This is the default. This option also creates source/lookup column joins when using the DB_LOOKUP function. This option is available when the Add Join Object if needed option is enabled.

Show adapter type

This option appears for Source and Target. When selected, the adapter type (for example, Fixed Format File or MS SQL Server) appears under the synonym name.

Note: On Add Source settings affect new sources that are added to a flow.

The On Add Target section has the following fields and options:

Automatically map all columns

Automatically maps all source columns to target columns of the same name.

Show adapter type

This option appears for Source and Target. When selected, the adapter type (for example, Fixed Format File or MS SQL Server) appears under the synonym name.

Data Flow Line Colors

Is a drop-down menu of available colors for data flow lines.

Reset colors

Restores the default colors.

Reference: User Preferences - Target Defaults

Target Defaults user preferences allow you to set default Prior to Load Values for each category of targets. You can also set commit levels for targets. All targets in the current and subsequent flows will use the commit level default set here.

On the Home tab, in the Tools group, click *Options*, and expand *Data Flow*. Then, click *Target Defaults* to access these settings.

The Target Defaults settings dialog box has two sections: Prior to Load Values and Flow Level.

The Prior to Load Values section has the following fields and options:

Relational

No changes

Does not delete the records already in a data target.

Delete all rows from table

Deletes all rows and then creates a database log.

Truncate table

Deletes all rows from the table but does not generate a database log. This is a faster option than Delete all rows from table because no database log is generated. If the target database does not support truncate, a delete is performed.

Focus

No changes

Does not delete the data target.

Drop Table

Deletes and recreates the data target.

Fixed Format File

No changes

Does not delete the data target.

Delete File

Deletes and recreates the data target.

The Flow Level section has the following fields and options:

Commit every row(s)

Specifies the number of rows to process before issuing a commit to a relational database target or a CHECK to a FOCUS target.

A value of zero (0) means that a COMMIT is only issued when the flow completes.

Note: This setting will be applied to all targets in a flow.

When several flows are loading multiple tables, it may be desirable to not issue a COMMIT until all the flows have completed successfully. In this case, a special value of NONE can be used to mean that Data Migrator should not issue any COMMIT commands. This means that it is the responsibility of the user to issue a COMMIT (or ROLLBACK) when appropriate. This can be done by creating a process flow that runs any required data flows and checks the return code for each data flow. When all flows have completed successfully, the process flow should then call a stored procedure that issues the COMMIT command, or if any data flow has failed, a ROLLBACK command.

This option should be used with caution. If no COMMIT or ROLLBACK is issued, the target tables could be locked.

Reference: User Preferences - Process Flow Designer

Process Flow Designer user preferences include using application directories.

On the Home tab, in the Tools group, click *Options*, and click *Process Flow Designer* to access these settings.

The Process Flow Designer settings dialog box has the following options:

Use application directory name with flow components

Includes the application directory name for objects in process flows.

Add flows to run as stored procedures

When flows are added to a process flow, the Run as stored procedure check box is automatically selected.

Show Notes

Allows you to add comments to process flow objects, directly from the workspace.

Process Flow Line Colors list box

Allows you to set colors for connector lines in process flows.

OK Default Condition. Changes the color of the OK connector in process flows.

Failed Default. Changes the color of the Failed connector in process flows.

Unconditional. Changes the color of connectors in data flows, and unconditional connectors in process flows.

Custom Line Color. Changes the color of the custom connector in process flows.

Reset colors

Restores the default colors.

Reference: User Preferences - Synonym Editor

Synonym Editor user preferences include using the application name with the synonym, supporting the extended option, and showing parent segments in the Join Editor.

On the Home tab, in the Tools group, click *Options*, and click *Synpnym Editor* to access these settings.

The Synonym Editor settings dialog box has the following fields and options:

Use application directory name with synonym

If selected, an application directory name will be used when selecting a synonym name for both referencing an existing synonym, and a transformation with db_lookup.

Undo/Redo Limit

Specifies the maximum number of undo operations allowed in the Synonym Editor.

Support extended options

If selected, the Sort Object and Style buttons appear in the Insert menu on the Metadata tab, in the Insert group.

Enable insert copy of existing synonym

If selected, this option will be available in the shortcut menu when a synonym or segment name is selected.

Insert child segment with snowflakes

If selected, when you insert a child segment (dimension), any of its child segments are inserted automatically.

Automatically detect new segment relations

When you create a new synonym in the Modeling View of Synonym Editor with the Automatically detect new segment relations option selected, and select or drag these three tables into the work area (modeling view), the relationships (joins) will automatically be created for you based on the foreign key information in the access file.

Automatically arrange segments/folders in Modeling View

If you select or drag multiple, or individual, tables while in Modeling View, they will be automatically arranged for you.

Show parent segments in Join Editor

If selected, this option controls whether or not columns in parent segments are displayed in the Join Editor for a cluster join, or for a synonym that references, or includes, other synonyms.

Automatically detect entry segment

Default Join Type

Allows you to select the default Join type. Options are Join All or Join Unique.

Auto Calculate Missing

This option controls the behavior when you insert a DEFINE field into a synonym where at least one operand in the expression has a MISSING-ON attribute. The available options are:

- **PROMPT.** Asks *Do you want to change MISSING to ON ALL"* whenever a new DEFINE is added. This is the default value.
- **ON.** Expressions are automatically adjusted so that the result of the expression is MISSING if any operand is missing.
- **OFF.** Expressions are not adjusted. The result of the expression is MISSING only if all of the operands are missing.

Modeling View Line Colors list box

Allows you to set colors for connector lines in the Modeling View.

Default. Changes the color lines in the Modeling View.

Highlighted. Changes the color of the highlighted lines in the Modeling View.

No keys. Changes the color of lines with no keys in the Modeling View.

Reset colors

Restores the default colors.

Reference: User Preferences - Run Options

Run Options user preferences include determining the number of rows to retrieve when sampling data and keeping your server connection alive.

On the Home tab, in the Tools group, click *Options*, and click *Run Options* to access these settings.

The Run Options settings dialog box has the following fields and options:

Maximum number of rows for test reports

Sets the number of rows retrieved to produce sample data when testing transformations or SQL. The default is 50.

Maximum number of columns for test reports

Sets the number of columns retrieved to produce sample data when testing transformations or SQL. The default is to retrieve all columns (with a highest value setting of 999999).

Test reports default format

Sets the format of reports for retrieving sample data when testing transformations, or SQL. The default is Default. There are seven report formats available:

- **Default.** Formats numeric and date columns based on edit options in the synonym.
- **HTML.** Produces the report in HTML format.
- **HTML - plain text.** Produces the report in plain text format.
- **Active Report.** Produces an HTML active report designed for offline analysis.
- **PDF.** Produces the report in PDF format.
- **Unformatted.** Does not apply formatting to numeric and date columns.

Omit missing (null) values in Data Profiling charts

Enable this option to generate Data Profiling bar and pie charts that omit missing values.

Stop processing if 0 rows selected

Allows you to stop processing if no data is returned. This setting is the default for new flows. You can override this option for an individual flow by resetting the execution properties of the flow.

Stop after DBMS errors

Allows you to stop processing the number of DBMS errors received if the number of errors specified is exceeded. This setting is the default for new flows. You can override this option for an individual flow by resetting the flow execution properties for that flow.

User Privileges

User privileges associated with a directory or other server objects, such as procedures or synonyms, can be managed from the DMC.

Note: These privileges can only be controlled by a user that is logged on to a secure server with a user ID with Server Administrator privileges. If you are logged on with an ID that is not a Server Administrator, you will only have access to view your own privileges. You cannot change them or view the privileges of another user.

Procedure: How to Access Privileges

There are two ways to access the Manage Privileges dialog box:

- Right-click an application directory and click *Privileges*.

  or

- Right-click an object (procedure, synonym, flow, and so on) and click *Privileges*.

Reference: Manage Privileges Dialog Box

To access the Manage Privileges dialog box, right-click an application directory, or an object, such as a procedure or synonym, and click *Privileges*.

The Manage Privileges dialog box has the following columns:

Subject

This column shows roles, users, and groups.

Roles

There are five roles that can be used for managing privileges. Additional roles can be added by the server administrator. The default roles are:

- **Server Administrator.** Has full control of the workspace, adapters, synonyms, and procedures.
- **Application Administrator.** Can create and run synonyms and procedures.
- **Server Operator.** Can start and stop the server, as well as monitor and kill agents.
- **Basic User.** Can execute procedures.
- **None.** Has no privileges.

Each role can have users or groups of users associated with it. These are shown indented under each role name.

Privileges

The privileges that are associated with each directory are:

- **Read.** The user can read the directory or file.
- **Write.** The user can write, create, edit, or delete the directory or file.
- **Execute.** The user can run procedures or access metadata.
- **List.** Lists files.

Inherit Subject

Shows if the privileges are inherited from those of a role, by showing that role.

Inherit Directory

Shows if the privileges are inherited from those of a directory, by showing that directory.

Description

An optional description for the directory or file.

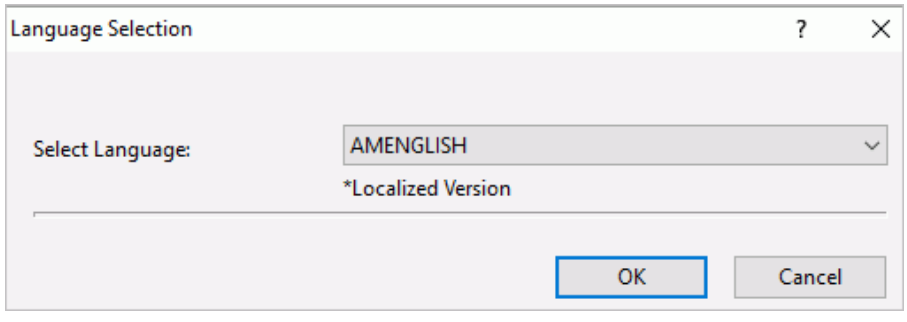
Configuring National Language Support

You can configure National Language Support (NLS) directly from the DMC. National Language Support enables the connector or server to translate both double-byte and single-byte National Character sets.

Procedure: How to Configure NLS

1. On the Home tab, in the Tools group, click *Language Selection*.

   The Language Selection dialog box opens, as shown in the following image.



   The Language Selection dialog box has the following fields and options:

   Select Language

   Controls the default value of the code page and the language of server error messages if they are available for translation. The default value is AMENGLISH (American English) on all platforms. Localized versions are noted by an asterisk (*).

2. Select a Language from the drop-down menu. (Note that localized versions are noted with an asterisk (*).)
3. Click *OK*.

Ending a Data Management Console Session

When you are finished working in Data Migrator, you should disconnect from the server and close the DMC.

Procedure: How to End Your Data Management Console Session

1. Save all changes.

2. Right-click the server name and select *Disconnect*.
3. Close the DMC by clicking the control button in the upper-right corner.

   or

   Selecting *Exit* from the Quick Access menu.

   or

   Double-clicking the Quick Access menu icon.

Note: When you close the DMC, you will be prompted to save changes if you did not already save them.

Chapter 5

Designing a Data Flow

A data flow contains the parameters for extracting information from data sources and loading it into data targets. You create data flows in the Data Management Console using the data flow workspace.

Typically, you build a data flow from a set of objects that identify data sources, data targets, data columns to include, and business rules governing how data is extracted, transformed, and loaded to form a data warehouse, data mart, or any other type of target.

You can join multiple data sources, filter and sort the extracted source data, and specify column mappings from the source to the target. You can also create specialized data flows, such as Direct Load and DBMS SQL flows.

This chapter describes how to design and create the different kinds of data flows.

In this chapter:

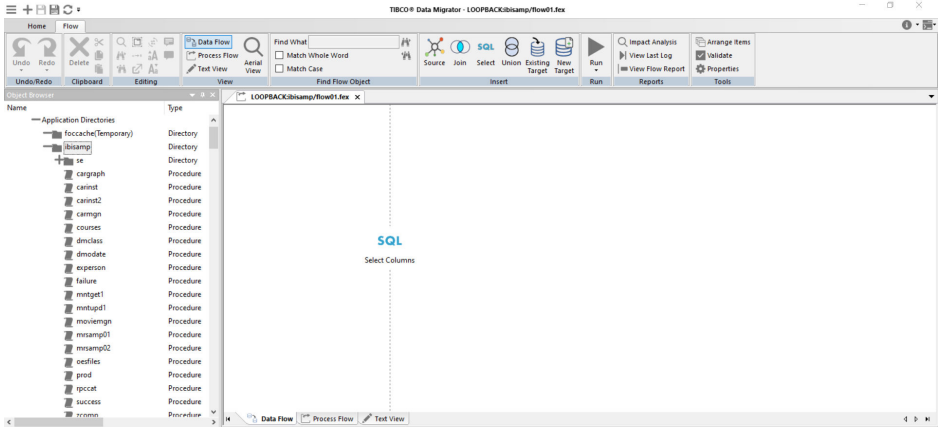
- The Data Flow Interface
- Simple Data Flow at a Glance
- Complex Data Flow at a Glance
- Creating a Data Flow
- Specifying Data Sources
- Creating a Source Transformation in a Data Source Object
- Joining Data Sources
- Adding a SELECT to a Data Flow
- Adding a UNION to a Data Flow
- Selecting Columns
- Testing the SQL Statement
- Specifying Filters in a Data Flow
- Specifying Sorts
- Selecting a Data Target
- Specifying Data Target Options
- Mapping Transformations
- Using Variables in a Flow
- Using Functions
- Column Formats and Mapping
- Setting Properties of a Flow
- Running a Data Flow
- Migrating Data Flows
- Creating a Direct Load Flow
- Creating a DBMS SQL Flow
- Creating a Quick Copy

The Data Flow Interface

The Data Flow interface consists of:

- The workspace, where you build your data flows to extract and move data from data source objects to data target objects. Arrows connect the objects and support the flow.
- Three tabs under the workspace, Data Flow, Process Flow, and Text View, which provide different views of the flow.
- The Flow tab in the ribbon, which enables you to set data flow properties, run flows, and arrange flow objects. The same tab is active when you create process flows.
- The navigation pane, where you choose and store flow components. It has six folders:
  - The Application Directories folder provides access to flows, procedures, and reports that can be generated from a flow. It also provides access to synonyms that describe data. You can use them as data sources or data targets in your flow.
  - The Adapters folder allows you to add and configure adapters.
  - The Resource Management folder is used with Resource Analyzer and Resource Governor.
  - The Workspace folder allows you to configure the server.

As shown in the following image, when you create a new flow, the Data Flow tab opens by default, with an SQL object in the workspace to anchor the flow. By default, the Flow tab is also active.



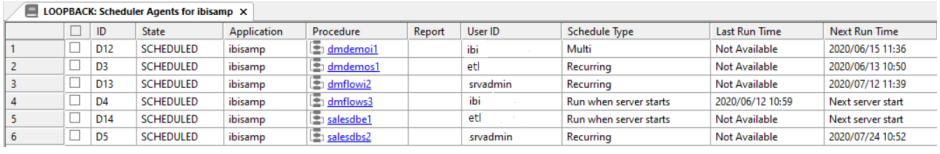
You can add data source objects and data target objects, and use the SQL object to establish rules for extracting and moving data between them.

For more information on the Data Flow Interface, see *Working in the Data Management Console* on page 103.

Procedure: How to Manage Scheduled Flows

1. Right-click *Application Directories* or an individual application directory, point to *Schedule and E-mail*, and then click *Scheduler Agents*.

   The Scheduler Agents for window opens, as shown in the following image.


LOOPBACK: Scheduler Agents for ibisamp

		ID	State	Application	Procedure	Report	User ID	Schedule Type	Last Run Time	Next Run Time
1	☐	D12	SCHEDULED	ibisamp	dmdemoi1		ibi	Multi	Not Available	2020/06/15 11:36
2	☐	D3	SCHEDULED	ibisamp	dmdemos1		etl	Recurring	Not Available	2020/06/13 10:50
3	☐	D13	SCHEDULED	ibisamp	dmflowi2		srvadmin	Recurring	Not Available	2020/07/12 11:39
4	☐	D4	SCHEDULED	ibisamp	dmflows3		ibi	Run when server starts	2020/06/12 10:59	Next server start
5	☐	D14	SCHEDULED	ibisamp	salesdbe1		etl	Run when server starts	Not Available	Next server start
6	☐	D5	SCHEDULED	ibisamp	salesdbs2		srvadmin	Recurring	Not Available	2020/07/24 10:52

2. Select or clear the check boxes for any flow you want to make active or inactive.
3. On the Report tab, from the Tools group, click *Activate Selected* or *Deactivate Selected*.

Reference: Advanced Filter

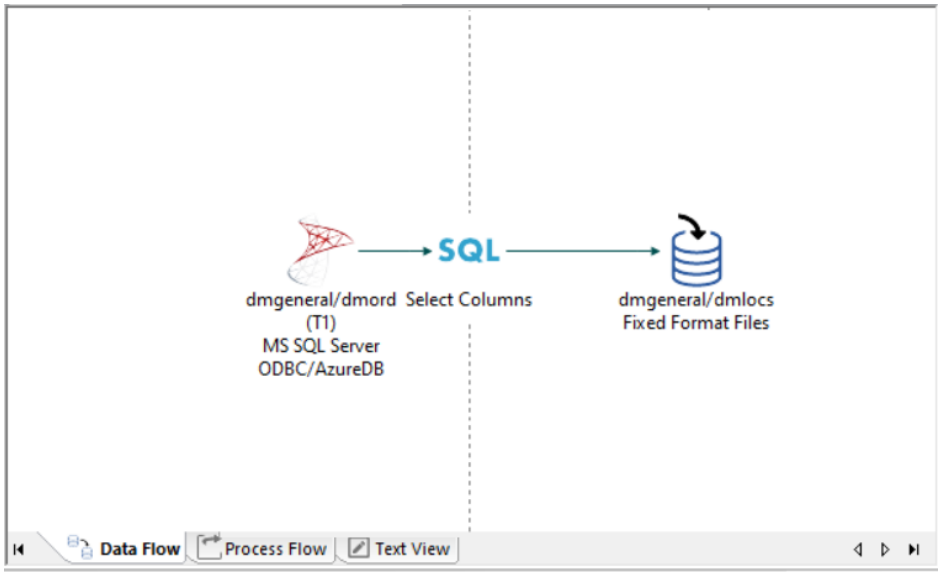
The Advanced Filter gives you finer control over the file types that appear in the navigation pane. For example, you could specify that you only want to see synonyms for the MS SQL Server tables. It also lets you restrict display to files with specified names (using wildcards) or descriptions, files modified in a specific time period, or those of a specified size.

On the Home tab, from the Filter group, click *Advanced*. The Filter Applications Tree dialog box opens.

Simple Data Flow at a Glance

You can use Data Migrator to build data flows ranging from the simple, where data is moved from one source into one target, to the complex, which employs multiple sources and targets and a variety of data extraction techniques.

This simple data flow consists of a data source, an SQL object, and a data target, as shown in the following image.



To create a simple data flow:

1. Open a new flow.

   The Data Flow tab opens by default, with an SQL object in the center of the workspace.

2. To add a data source object, drag a data source object from the list of synonyms in the navigation pane into the workspace, to the left of the SQL object.

3. Select the columns that you want to extract from the source and copy into the target.

   Note: If *Automatically select all columns* is selected in the Data Flow Designer user preferences tree from the Options dialog box, you do not need to manually select columns.

4. To add a data target object to the data flow, right-click the workspace to the right of the SQL object, select *Add Target*, and then click *New*.

5. Save and submit the data flow.

   Note: Flow names can be up to 64 characters.

6. View the log file for data flow details.

Reference: **Data Flow Objects in the Workspace**

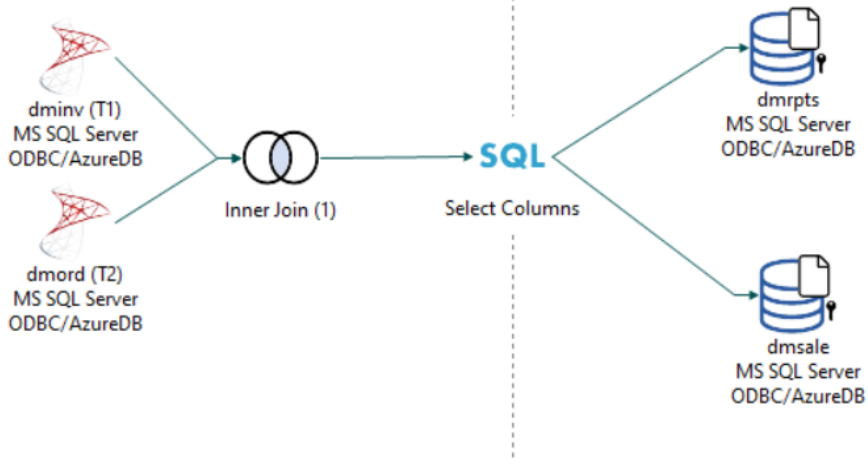
Data flow objects appear as follows in the workspace:

Object	Function
	Data source - *sourcename* (Tn) or *appdir/sourcename* (Tn) Indicates a data source for the data flow, where *sourcename* is the name of the synonym and Tn is the table number. A key icon in this object indicates that the synonym has a key identified. **Note:** The source icons are database-specific. The one shown is for a Microsoft SQL Server source.
	SQL/Select Columns Contains the generated SQL used to extract the data, based on selected columns, filters, and sorts used in the data flow. A filter, or sort, icon in this object indicates when a filter or sort has been specified.
	Target - *targetname* or *appdir/targetname* Indicates a target for the data flow, where *targetname* is the name of the synonym. A key icon indicates if the target table has key columns identified. A red star on the icon indicates a new target table.

Complex Data Flow at a Glance

You can define a complex data flow that extracts information from multiple data sources, and copies it into multiple data targets using extract transformations, joins, filters, and sorts to refine the results.

This data flow consists of two data sources, an extract transformation, a join, a filter, a sort, an SQL object, and two data targets, as shown in the following image.



To create a complex data flow:

1. Open a new flow.

   The Data Flow tab opens by default, with an SQL object in the center of the workspace.

2. To add a data source, drag a data source object from the navigation pane into the workspace, to the left of the SQL object.

3. Add a second data source to the flow.

   The data sources are connected by a join object, which is linked to the SQL object.

   Note: The default setting of the DMC automatically adds a join object to the workspace. If the DMC is not automatically adding join objects, you can change the setting by selecting *Add Join Object if needed* in the Data Flow Designer user preferences found in the Options dialog box.

4. Build an expression that creates a new column for extraction in the second data source.

5. Specify join properties.

   An inner join is the default. If you are using the default setting for the DMC, an inner join is created automatically for columns that are identically named and formatted, if one of them is a key column.

   Note: If an inner join is not automatically created, you can change the setting by selecting *Add Join Object if needed* in the Data Flow Designer user preferences found in the Options dialog box.

6. Create an expression that filters the data prior to extraction.
7. Select the columns you want to extract from the source and copy into the target.

   Tip: You do not need to manually select columns if *Automatically select all columns* is selected in the Data Flow Designer user preferences found in the Options dialog box.

8. To add two data target objects to the data flow, drag one from the navigation pane and create the other.
9. Map the columns from the joined data sources to the existing data target.

   The columns in the new data target are mapped automatically.

   Tip: You do not need to map columns manually if the source and target column names are the same if *Automatically map all columns* is selected in the Data Flow Designer user preferences found in the Options dialog box,

10. Save and run the data flow.

    Note: Flow names can be up to 64 characters.

11. View the log file for data flow details.

Tip: You can create this data flow by completing *Exercise 1: Creating the Data Flow* on page 419. In addition, the exercises in *Advanced Tips and Techniques* on page 541, include several complex data flows that illustrate a range of useful options.

Creating a Data Flow

The first step in building a data flow is to create an empty flow. You will then be able to add objects to this flow, and set up mappings and relationships.

Procedure: How to Create a Data Flow

1. Right-click the application directory where you want the new data flow.
2. Point to *New* and then click *Flow*.

   or

   Click the New button on the Quick Access Toolbar.

An empty data flow appears in the data flow workspace. The Data Flow tab opens by default, with an SQL object in the center of the workspace.

Adding and Deleting Objects in Your Data Flow

You can add or delete objects in the data flow workspace. Objects on the left of the SQL object are sources, and objects on the right are targets.

Note: When adding objects, there is no limit to the area they can take up. If the number and arrangement of objects exceeds the viewable area of the workspace, a scroll bar will appear.

Procedure: How to Add Objects to Your Data Flow

- ❑ Drag objects from the navigation pane into the data flow workspace.

  or

- ❑ Right-click the data flow workspace. On the left side of the SQL object, you can click *Add Source*, *Add Join*, *Add Select*, or *Add Union*. On the right side of the SQL object, you can click *Add Target*.

  or

- ❑ Right-click a *Join*, *SQL*, or *Union* object and click *Add*. Select the object from the available list.

  or

- ❑ On the Flow tab, in the Insert group, click one of the commands, and drag it into the workspace.

  If you are adding a source or an existing target, the Select Synonym dialog box opens.

  Select a synonym and click the *Select* button.

Procedure: How to Search for Existing Synonyms to Use as Sources or Targets

The Select Synonym dialog box has a search feature that allows you to find synonyms by searching their names and descriptions.

1. Right-click the workspace while working with a data flow, point to *Add Source* or *Add Target*, and then click *Existing*.

   The Select Synonym dialog box opens.

2. Click the *Find* button at the upper right.

   The Find dialog box opens.

3. Type a string in the *Find What* field, and click the *Find Next* button. You can refine your search with match whole word only and match case. You can also search in either direction.

   The first synonym name, or description containing the string, is highlighted in the Select Synonym dialog box.

4. Click the *Find Next* button to find the next match.

Note: After closing the Find dialog box, you can still search for the next match by pressing the F3 key. Shift + F3 will find the next match searching upward.

Reference: Select Synonym Dialog Box

To open the Select Synonym dialog box while working in the Data Flow tab, right-click the workspace and click *Add Source* or point to *Add Target*, and then click *Existing*.

The Select Synonym dialog box has five default columns, but this can be customized. The following fields and options are available:

Look in

Is the name of the application directory where the synonyms are located. Select the server name to see all synonyms in all directories.

Name

Is the name of the synonym.

Type

Is the type of synonym.

Description

Is the description of the synonym.

The following columns are available if you are adding a source to a Join object that already has a synonym selected:

Condition

Are the join conditions that can be used to join the already selected synonym to a new synonym.

`Confidence`

Shows the confidence level, based on matching column names and data types for each prospective new source, showing how compatible it is with the existing source.

`Filtering`

Enables you to filter on sources that satisfy the following conditions when adding a source to a join:

- **Foreign to Primary.** Lists sources that have a foreign key relationship to a primary key in the new table. This option is checked by default.
- **Matching names.** Lists sources that have column names that match key column names in the existing source. This option is checked by default.
- **Primary to Foreign.** Lists sources that have a foreign key relationship to a primary key in the already selected table.
- **Fuzzy matching names.** Lists sources that have column names that are similar to key column names in the existing source.
- **Other (or unmatched).** Lists sources that do not have columns that satisfy the other conditions.

New 

Allows you to create a new synonym.

Find 

Allows you to search the application directories for a particular synonym.

Name text box

Allows you to enter a synonym name, or shows the name you have selected.

Type

Allows you to select the synonyms by their adapter type. For example, you can display only synonyms for MS SQL Server tables.

Select button

Adds the highlighted synonym to the workspace.

Title Bar

Left-click a column header in the title bar to sort the list of synonyms on that column. Do it again to reverse the order. Right-click a column header in the title bar to open the context menu with the following options:

- **Sort Ascending/Descending.** Sorts the list of synonyms on the selected column.
- **Find.** Opens the search dialog.
- **Tree View/List View.** Switches views of the synonyms.
- **Best Fit.** Optimizes the column widths for best viewing.
- **Customize.** Allows you to select the columns that are displayed,

Procedure: How to Delete Objects From Your Data Flow

Right-click the object you want to remove and click *Delete*.

Connecting and Disconnecting Objects in Your Data Flow

The connections between objects in the data flow workspace allow information to flow between them.

The DMC default User Preferences automatically connects objects.

- Objects dragged onto the left (source) side of the workspace are connected to the SQL object, or, if more than one data source is dragged onto the workspace, to each other through a Join object.
- Objects dragged to the right (target) side of the workspace are connected to the SQL object.

Note: The Options dialog box provides access to the DMC User Preferences. If the *Add Join Object if needed* setting on the *Data Flow Designer* page is selected, Source objects are automatically connected to a Join object, if more than one is dragged onto the workspace. For details, see *How to Automatically Add Join Objects* on page 205.

Procedure: How to Connect Objects

When objects are deleted, you may need to recreate the connections between some of the remaining objects.

1. Right-click an object, hold the mouse button down, and drag to the object you want connected.

2. Release the mouse button.

An arrow will appear indicating that a connection has been made.

You may also need to create connections if the DMC default settings have been changed.

Procedure: How to Disconnect Objects

Right-click the arrow connecting two objects, and click *Delete*.

or

Right-click an object, hold the mouse button down, and drag an arrow to a different object.

Saving a Data Flow

Data flows are saved on the Server. The first time you save a data flow, you will select an application directory and assign a name to it. After that, you can save it with the same name, or assign a new one.

The following restrictions apply to flow names:

- Names can contain only letters, numbers, and underscores.
- Uppercase letters are converted to lowercase when the flow is saved.
- Names can be up to 64 characters long, except on z/OS with PDS deployment, where names are limited to 8 characters.

Procedure: How to Save a Data Flow

1. Click the Save button on the Quick Access Toolbar.

   Note: A flow can only be saved once changes have been made to it. If no changes have been made, both the button and Save option in the main menu will be inactive (grey).

   If this is the first time a flow is being saved, the Save As dialog box will open. See *How to Save a Data Flow with a New Name* on page 182.

2. Check the console window for a message indicating that the flow was saved successfully.

Procedure: How to Save a Data Flow with a New Name

1. Click the Save As button on the Quick Access Toolbar.

   The Save Procedure As dialog box opens.

2. Select the application directory where you want to save the flow.
3. In the Name box, enter the name you wish to use for the flow.

Click *Save*.

4. Check the console window for a message indicating that the flow was saved successfully.

Specifying Data Sources

Once you have opened a new data flow, you need to identify the data sources you want to use, and add them to the flow. Data sources are represented by synonyms in the navigation pane, which are under an application directory folder when the Synonyms button is selected. The options available when you right-click a synonym are described in *Synonyms Shortcut Menu Options* on page 118. For guidelines on synonyms used as data sources, see *Using Synonyms for Data Sources and Data Targets* on page 88.

When creating a data flow, information about the data source, such as the column names, is required and supplied when the Server is connected.

When creating or editing a data flow, in order to test your flow, data sources referenced in the flow must be available.

- The server with a relational data source must be running to use that source.
- A data set must be allocated for data in a fixed format file or VSAM structure.
- The source servers must be running. and they must be reachable from the Hub server in a typical configuration with a Hub server and one or more remote servers.

Procedure: How to Determine What is in a Data Source

Before you add a data source to a flow, view the columns to verify that it contains the data that you want and that the data is available.

To view a sample of the data that a synonym will produce, right-click the synonym in the navigation pane and click *Sample Data*. The sample data will appear in the data flow workspace, as shown in the following image.

LOOPBACK: Data for ibisamp/dminv. Limit 50 rows.

	Product Number	Product Name	Quantity in Stock	Sale Price	Our Cost	Product Category	Product Type
1	1004	2 Hd VCR LCD Menu	43,068	179.00	129.00	VCRs	Analog
2	1006	Combo Player - 4 Hd VCR + DVD	13,527	399.00	289.00	DVD	Digital
3	1008	DVD Upgrade Unit for Cent. VCR	199	199.00	139.00	DVD	Digital
4	1010	750SL Digital Camcorder 300 X	10,758	999.00	750.00	Camcorders	Digital
5	1012	650DL Digital Camcorder 150 X	2,972	899.00	710.00	Camcorders	Digital
6	1014	340SX Digital Camera 65K P	990	249.00	199.00	Cameras	Digital
7	1016	330DX Digital Camera 1024K P	12,707	279.00	199.00	Cameras	Digital
8	1018	250 8MM Camcorder 40 X	60,073	399.00	320.00	Camcorders	Analog
9	1020	150 8MM Camcorder 20 X	5,961	319.00	240.00	Camcorders	Analog
10	1022	120 VHS-C Camcorder 40 X	2,300	399.00	259.00	Camcorders	Analog
11	1024	110 VHS-C Camcorder 20 X	4,000	349.00	249.00	Camcorders	Analog
12	1026	AR3 35MM Camera 10 X	12,444	129.00	95.00	Cameras	Analog
13	1028	AR2 35MM Camera 8 X	11,499	109.00	79.00	Cameras	Analog
14	1030	QX Portable CD Player	22,000	169.00	99.00	CD Players	Digital
15	1032	R5 Micro Digital Tape Recorder	1,990	89.00	69.00	Digital Tape Recorders	Digital
16	1034	ZT Digital PDA - Commercial	21,000	499.00	349.00	PDA Devices	Digital
17	1036	ZC Digital PDA - Standard	33,000	299.00	249.00	PDA Devices	Digital

Note: After a synonym has been dragged into the data flow workspace, you can view its columns, sample data, and transformations. Right-click the synonym and click *Expand*. You can view the data by clicking either the *Columns*, *Sample Data*, or *Test Transforms* tab.

Procedure: How to Add a Data Source to a Data Flow

1. If you have not already created a new data flow, right-click the application directory where you want the new data flow, point to *New*, and then click *Flow*.

   An empty data flow appears in the data flow workspace. The Data Flow tab opens by default, with an SQL object in the center of the workspace.

2. Drag a data source object from the navigation pane into the data flow workspace, to the left of the SQL object.

   or

   Right-click in the workspace to the left of the SQL object, and click *Add Source*.

   The Select Synonym dialog box opens.

3. Select a synonym and click the *Select* button.

Procedure: How to Change a Data Source in a Data Flow

With the data flow open in the workspace:

1. Right-click the Source object that you want to change and click *Change Source*.

The Select Synonym dialog box opens.

2. Select the replacement synonym and click the *Select* button.

Note: If you change a data source that has source transformations associated with it, the transformations are preserved, but they will appear red if the new source does not contain all of the columns used in the transformation.

Procedure: How to Use a Relational Stored Procedure as a Data Source

You can use a relational stored procedure as a data source in a flow if you have created a synonym for that stored procedure. For information on creating a synonym, see *How to Create a Synonym for a Relational Stored Procedure* on page 57.

One use for this would be to perform a lookup, for example, passing parameter values to the stored procedure and retrieving an answer set based on the supplied parameter.

This is supported for:

- Db2 (using CLI)
- Oracle (using CLI)
- SQL Server (using OLE DB and JDBC)
- Sybase IQ (using ODBC).
- Teradata (using ODBC)
- MySQL (using JDBC)

If you have created a synonym for a relational stored procedure, it will appear in the navigation pane, along with all other synonyms.

As an example, we are using an Oracle stored procedure that has empno as a parameter, and that retrieves information associated with that employee number. The Oracle stored procedure goes against the EMP Oracle sample table. We will join the stored procedure to a fixed format file that contains valid employee numbers.

The sample stored procedure is shown below.

```
/* create get_emp routine */

CREATE OR REPLACE PACKAGE get_emp AUTHID CURRENT_USER AS
PROCEDURE get_emp_proc (
   pempno     IN  scott.emp.empno%TYPE,
   pename     OUT scott.emp.ename%TYPE,
   pjob       OUT scott.emp.job%TYPE,
   pMgr       OUT scott.emp.mgr%TYPE
);
END get_emp;
```

```
/
show err
CREATE OR REPLACE PACKAGE BODY get_emp AS
PROCEDURE get_emp_proc (
   pempno     IN  scott.emp.empno%TYPE,
   pename     OUT scott.emp.ename%TYPE,
   pjob       OUT scott.emp.job%TYPE,
   pMgr       OUT scott.emp.mgr%TYPE
)

IS

CURSOR gRec_cur IS
   SELECT  ename,
      job,
        mgr
   FROM scott.emp
   WHERE EMPNO = pempno;
gRec_rec gRec_cur%ROWTYPE;
BEGIN

  OPEN gRec_cur;

  FETCH gRec_cur INTO gRec_rec;

  IF (gRec_cur%FOUND) THEN
     pename := gRec_rec.ename ;
     pjob   := gRec_rec.job ;
     pmgr   := gRec_rec.mgr ;
  END IF;
  CLOSE gRec_cur;

  EXCEPTION
     WHEN OTHERS THEN
     ROLLBACK ;
     RAISE;
END get_emp_proc;
END get_emp ;
/
show error

GRANT EXECUTE ON get_emp TO public ;
```

To use a relational stored procedure as a data source:

1. With a data flow open in the workspace, drag the synonym into the workspace, to the left of the SQL object.

Note: The flow we are using joins a fixed format file named ora_emps with the get_emp_get_emp_proc Oracle stored procedure. The fixed format file has only one column, empno, which is joined to the stored procedure. For each empno in the fixed format file, the stored procedure runs and retrieves the name, job, and manager information for that empno.

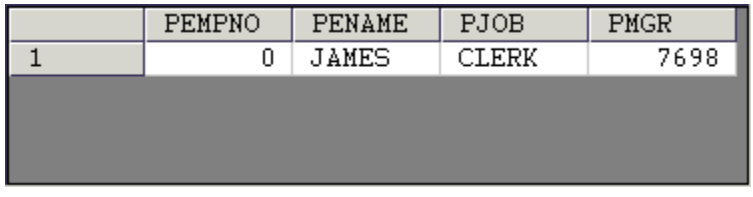
2. Right-click the synonym, point to *Operations*, and then click *Sample Data*.

   The Sample Data dialog box opens.

3. Select the parameter check boxes, enter their values, and click *Sample Data*.

   Note: If the stored procedure can have more than one answer set, you need to select the appropriate one.

   The sample data appears in the workspace.



	PEMPNO	PENAME	PJOB	PMGR
1	0	JAMES	CLERK	7698

Whenever you test sample data, you will need to supply the values for the parameters.

Reference: Data Source Object Options

After a synonym has been dragged into the workspace, the following options are available when you right-click the synonym:

Expand/Collapse

Opens or closes a window that allows you to view the columns, sample data, and transformations of the synonym.

Refresh Columns

Updates the list of columns in a synonym.

Source Transformations

Opens the Source Transformations window. For details, see *Creating a Source Transformation in a Data Source Object* on page 190

Operations

Open

Opens the synonym in the workspace. The file, segments, and columns appear on the left and their attributes and values appear on the right.

Edit as Text

Opens the synonym syntax in the workspace, where it can be edited.

Sample Data

Shows a sample of the data associated with the synonym.

Data Profiling

Provides the data characteristics for synonym columns.

Change Source

Opens the Select Synonym dialog box, where a different source can be selected.

Delete

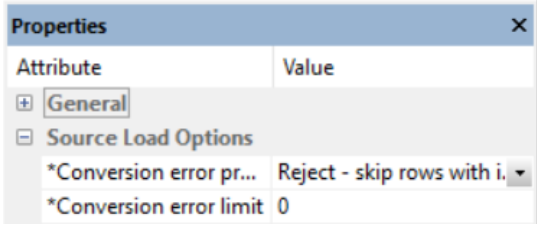
Deletes the synonym from the data flow.

Properties

Opens the attributes of the synonym. The following General properties are available:

- **Adapter.** The name of the adapter.
- **Application Directory.** The application directory for the synonym.
- **Synonym Name.** The name of the synonym.
- **Description.** A description of the selected synonym.
- **Table Name.** The name of the source table or data file.
- **Connection.** The name of the connection.
- **Tag.** The correlation name of the synonym. You can change the default name (T1, T2, and so on) to one that is more descriptive.
- **Display Name.** An alternate name you can give to the data source.
- **Notes.** An annotation describing the synonym.

A format conversion error occurs when a field described with a numeric USAGE format contains character values. When the source is a delimited or Excel File, the following Source Load Options are available to control the processing of format conversion errors, as shown in the following image:



- **Conversion error processing.** The default value, *Reject*, rejects the record. The value *Ignore* replaces the values causing errors with default values (null if the column is nullable, zero otherwise), and continues processing the record.
- **Conversion error limit.** Sets a limit, up to 999999, for the number of records with warnings about data conversion errors. The default value (0) allows unlimited warning messages. When the limit is reached, processing terminated and the system variable FOCERRNUM is set to 1346.

When more than one data flow object is selected, the following additional options are available:

Collapse Selected

Collapses the selected objects. This option is only available when one or more of the selected objects is expanded.

Delete

Deletes the selected objects. This option is not available when the SQL object is selected.

Updating the List of Data Sources

You can update the list of data sources by refreshing the tree. You can update the list of columns in a data source by refreshing the source.

This is useful when data sources are added to the server, or when columns are added to a data source while you are building a data flow, and you want these changes reflected in the data source or column list.

Procedure: How to Update the List of Data Sources

In the navigation pane, right-click the server, the Applications Directories folder, or subfolder, and click *Refresh*.

Procedure: How to Update the List of Columns

In the navigation pane, right-click a data source, and click *Refresh*.

Note: After you drag the data source onto the data flow workspace, you can refresh the column list by right-clicking the data source object and clicking *Refresh Columns*.

Creating a Source Transformation in a Data Source Object

You can create a source transformation in a data source to add intermediate calculations to the request. Source transformations are performed on a data source object as the records are read, before any filtering or aggregation is performed. This is often useful for complex calculations that are composed of multiple expressions. Creating a source transformation is also helpful in performing data type conversions when multiple steps are required, and if you want to aggregate or filter a calculated column.

Source transformations are only performed for a particular data flow.

A source transformation can contain an expression or a constant.

- If the source transformation contains an expression, you can create the expression with the Transforms Calculator.
- You can also enter a value or expression in the expression area. The format is automatically determined and displayed.

Note:

- If you are adding a source transformation to a hierarchical file (for example, FOCUS/FDS), the WITH attribute may be required. The WITH attribute may also be required if you add a source transformation that references a constant.
- Use of some source transformations disables Automatic Passthru. For more information, see *Using Automatic Passthru* on page 476.
- You can create a virtual column that appears whenever a data source is used by adding it to the data source itself via the navigation pane. For more information, see *Adding Virtual Columns (DEFINE) in a Synonym* on page 527.
- When re-using a synonym in a data flow, such as for a recursive join, the transformations must be the same in all instances of the synonym in the flow.

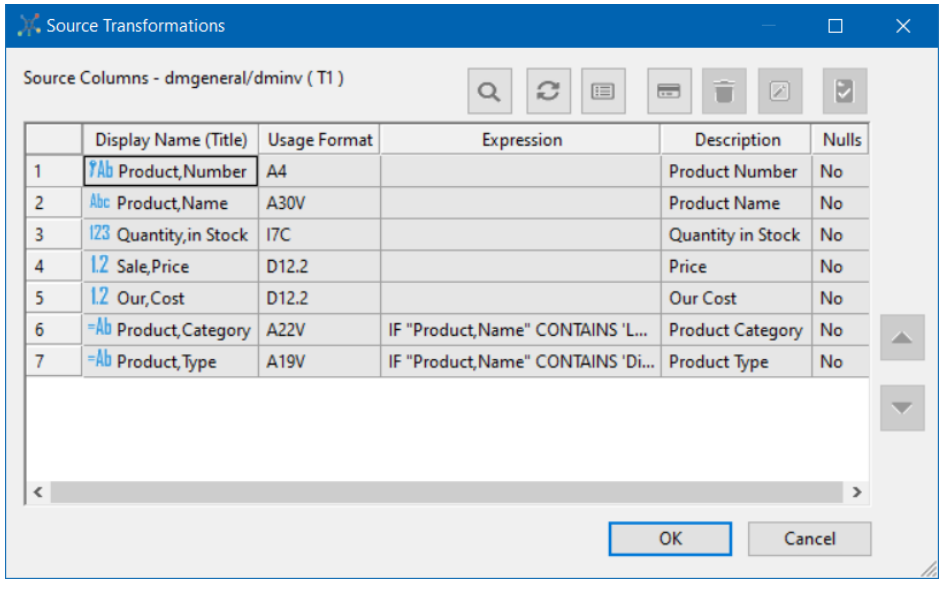
  If different transformations are required in each instance, you should create multiple synonyms with different synonym names for these sources and use them in the data flow.

In addition to source transformations, you can use the Relational Expression tab of the Transformation Calculator to create a relational expression transformation that can be used in a filter.

Procedure: How to Create a Source Transformation in a Data Source Object

1. From the data flow workspace, double-click a data source object, or right-click it and click *Source Transformations*.

   The Source Transformations dialog box displays all of the columns in the data source.



   Note: The columns of the synonym are shaded because they cannot be edited here. Only source transformations can be edited in this dialog box.

2. Click the Insert Transforms button  .

   The Transformation Calculator opens.

3. Type a name and format for the virtual column.

   Note: If you do not supply a format, the DMC will automatically provide the format based on the expression. Similarly, if you supply the wrong format, the DMC will replace it with the correct format using a pop-up notification to advise you of the change.

4. Use the Columns/Variables and Functions tabs, and the calculator buttons, to build the expression for the virtual column. For information about using functions, see *Using Functions* on page 325. For information about using variables, see *Using Variables in a Flow* on page 314.

Note: When you create expressions, keep in mind their order of evaluation. For details, see *Order of Evaluation* on page 203.

5. Click *OK*.

   The new column appears in the Columns list.

6. To verify that the transformation is working properly, click the Test Transforms button .

   The Test Transformations dialog box opens, showing the result of the source transformation.

7. Click *OK* to close the Source Transformations dialog box.

Procedure: How to Create a Transformation That Uses a Variable

From the Source Transformations dialog box:

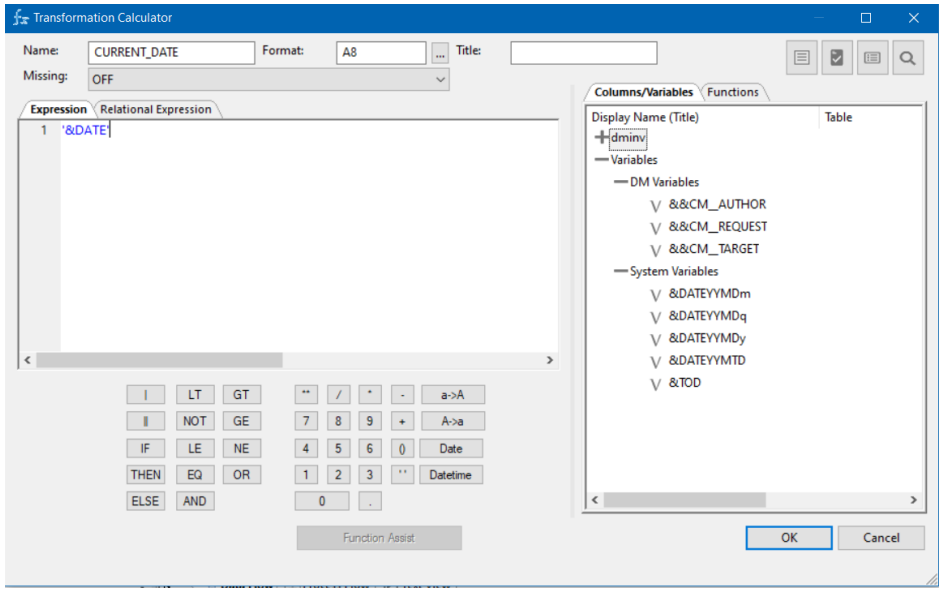
1. Click the Insert Transforms button.

   The Transformation Calculator opens.

2. Type a name and type or select a format for the virtual column.

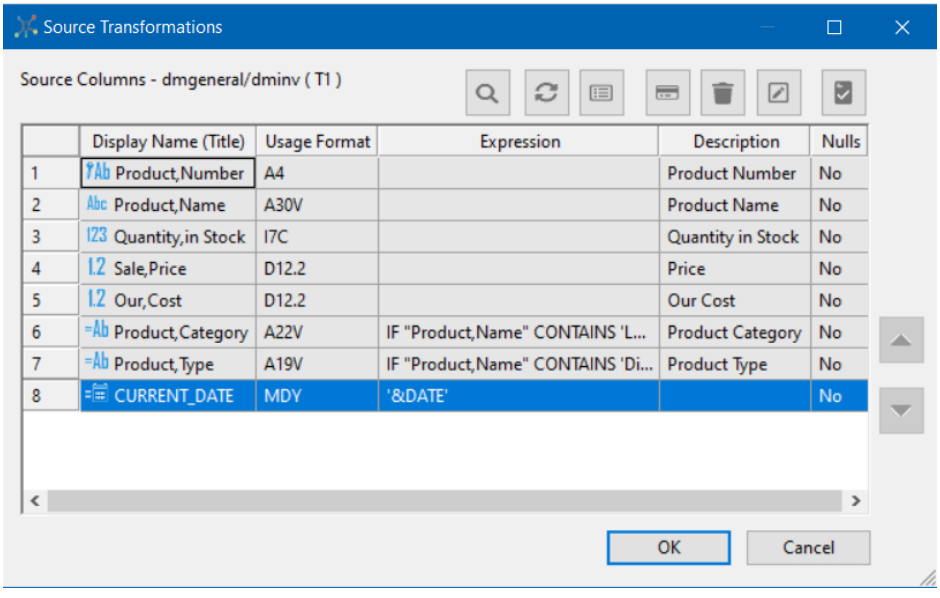
   Note: If you do not supply a format, the DMC will automatically provide the format based on the expression. Similarly, if you supply the wrong format, the DMC will replace it with the correct format using a pop-up notification to advise you of the change.

3. Double-click the variable you want assigned to the column from the Columns/Variables tab. You can also type in the variable, which must be enclosed in single quotation marks (‘), as shown in the following image.



4. Click *OK*.

The variable appears in the Expression column in the Source Transformations dialog box, as shown in the following image.



For more information about using variables, see *Using Variables in a Flow* on page 314.

Procedure: How to Create a Relational Expression in the Transformation Calculator

The Relational Expression tab of the Transformation Calculator allows you to build an expression that can be used as a filter.

From the Source Transformations dialog box:

1. Click the Insert Transforms button.

   The Transformation Calculator opens.

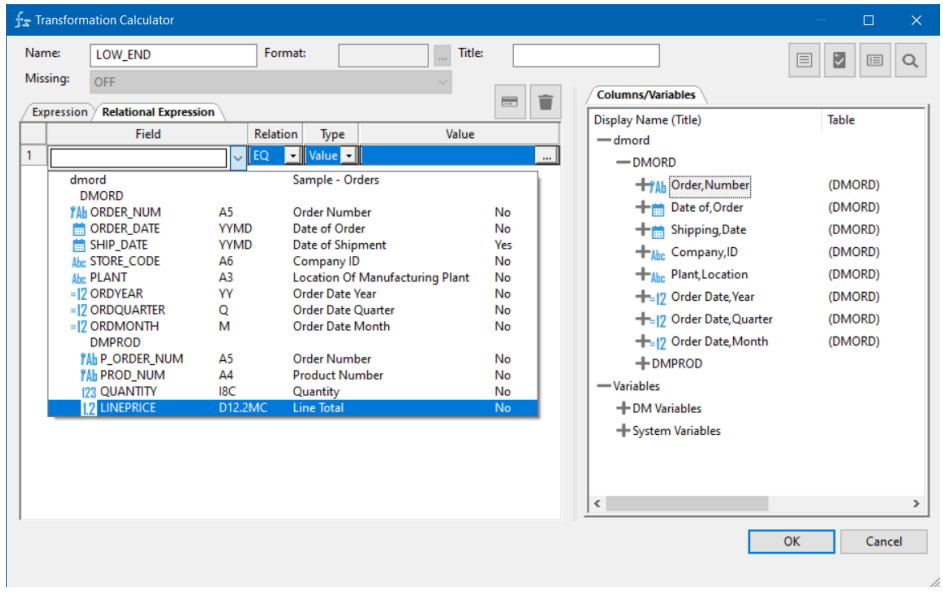
2. Click the *Relational Expression* tab.
3. Type in a name for the column.

   Note: Since the transformation will always evaluate to either 1 (true) or 0 (false), the format will be automatically set to I1.

4. Optionally, type in a title for the column.
5. Click the Add New Row button above the Expression field.

   A blank line is added to the Expression field.

6. Click the drop-down menu in the Field column, as shown in the following image.



7. Select a Column from the drop-down menu.
8. Select a Relation from the drop-down menu.
9. Select a Type from the drop-down menu.

   The following options are available:

   - ❏ Value
   - ❏ Field
   - ❏ Variable

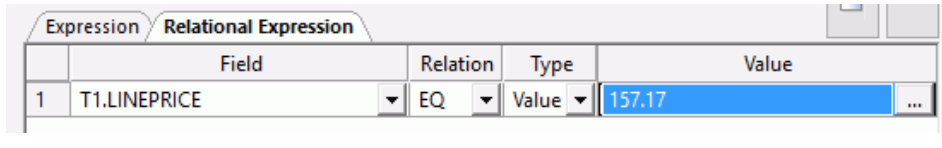
   Note: This field is not available if the Relation is set to IS MISSING or IS-NOT MISSING. If the Relation is set to DB_FILE or NOT DB_FILE, Type must be a value.

10. Type in a value or click the ellipsis (...) to select an available value.

    The Value(s) Selection dialog box opens.

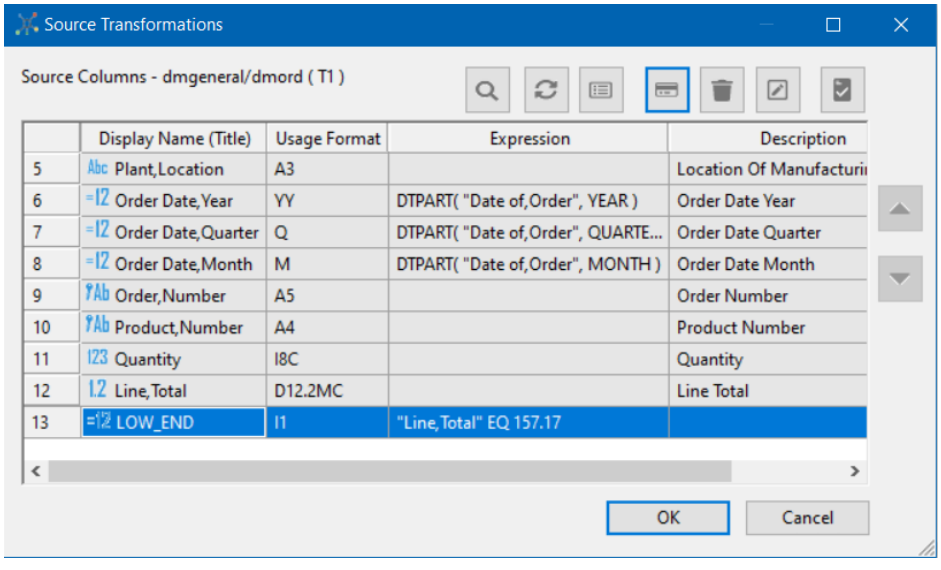
11. Select one (or more) Available Values and move it to the Selected Value(s) pane using the arrow. Click *OK*.

    The Transformation Calculator should look like the image below.



12. Click *OK* to close the Transformation Calculator.

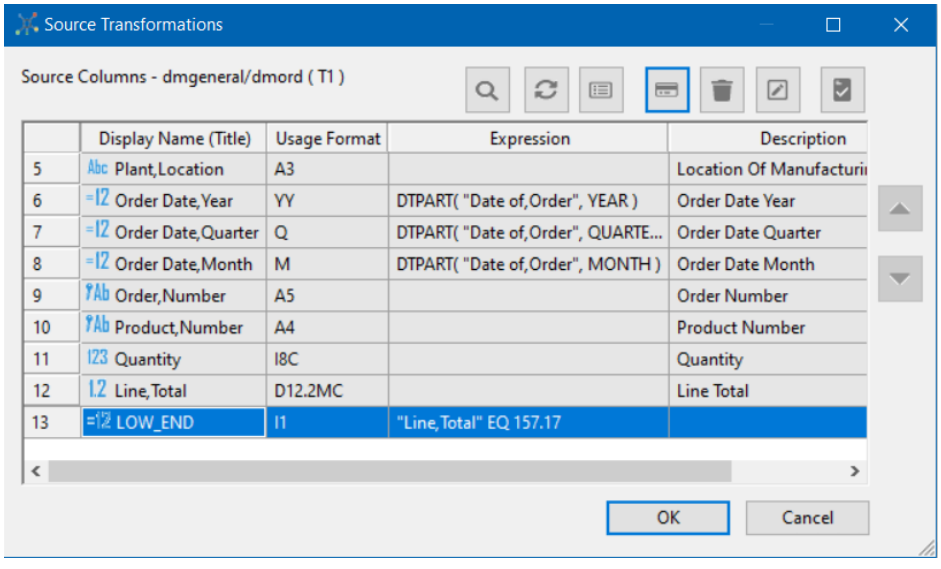
The relational expression appears as a column in the Source Transformations dialog box, as shown in the following image.



Note: Using a source transformation column in a filter or as a selected column may disable Automatic Passthru, so it should be used with care.

Reference: Source Transformations Dialog Box

To access the Source Transformations dialog box from the data flow workspace, double-click a source, or right-click a source and click *Source Transformations*.



The Source Transformations dialog box has the following fields and options:

Source Columns

Lists the columns in the data source. The first section contains the column order.

Display Name (Title)

Is the name of the column, and also shows a symbol that indicates regular columns, key columns, index columns, and virtual columns.

Format

Is the format of the column.

Expression

Is the expression for the column.

Description

Is the description of the column.

Find

Allows you to search for a column.



Refresh

Refreshes the columns in the data source.



Sample data

Shows a sample of the data from the data source.



Insert transforms

Opens the Transformation Calculator to create a transformation.



Delete transforms

Deletes the selected transformation.



Edit transforms

Opens the Transformation Calculator to edit a transformation.



Test transforms

Tests the transformations.

Up and down arrows

Moves the selected transformation up or down within the Source Transformations list.

Reference: Source Transformations Shortcut Menu

To access the source transformations shortcut menu, right-click any column of a transformation.

The source transformations shortcut menu has the following options:

Edit Transformation (Advanced)

Opens the Transformation Calculator for the selected transformation. This is the default action.

Insert Transformation

Displays a shortcut menu with a list of available functions that can be added as a transformation. Selecting one of the functions opens the Function Assist dialog box, which allows you to specify parameters for that function.

Delete

Deletes the transformation.

Test Transformations

Tests all of the transformations. This can also be done by clicking the *Test Transforms* button.

Error Details

This option is only available if the selected transformation contains an error, which is displayed in red text. Selecting this option opens a dialog box with the error code and a short explanation of the error.

Copy Transformations

Copies the selected transformations to memory, so that they can be pasted into this, or another, flow.

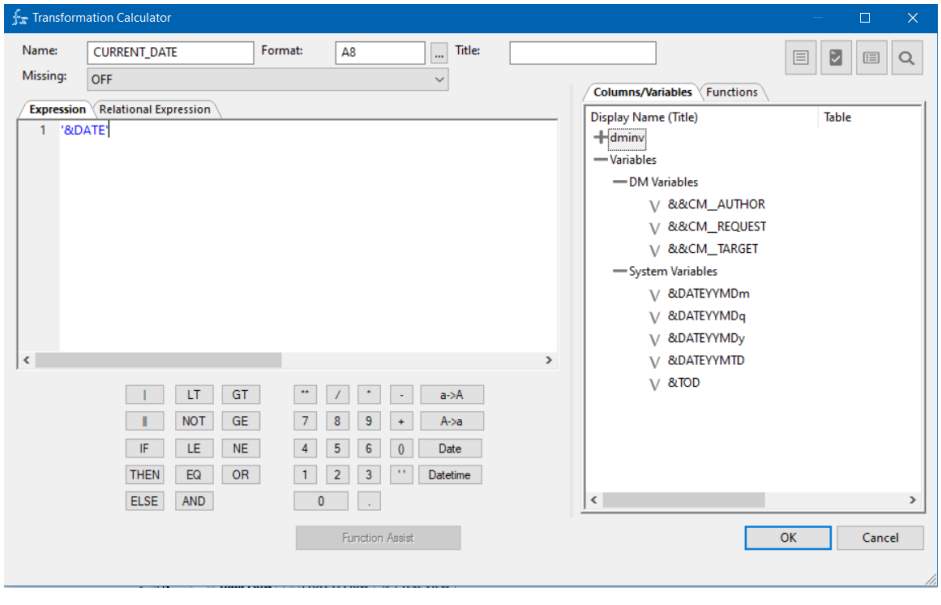
Paste Transformations

Pastes transformations from memory into the transformation list.

Note: Copying and pasting transformations does not use the Windows clipboard.

Reference: Transformation Calculator

To access the Transformation Calculator, click the *Insert* button in the Transformations dialog box.



The Transformation Calculator has the following fields and options:

Name

Is the name of the column.

Format

Is the format of the column.

Note: If you do not supply a format, the DMC will automatically provide the format based on the expression. Similarly, if you supply the wrong format, the DMC will replace it with the correct format using a notification to advise you of the change.

Title

Is the title of a new target table created by a data flow. The title will be included in the synonym that is created.

Missing

Indicates if missing (null) values are allowed in this field. The following options are available:

- **OFF.** The result is never missing.
- **ON.** The result is missing only if all missable fields are missing.
- **ON ALL.** The result is missing if any missable fields are missing.

Expression tab

Displays the expression dialog box.

Relational Expression tab

Displays the Relational Expression Builder dialog box.

Columns/Variables tab

Displays available columns and variables in a hierarchical tree or in a grid.

Functions tab

Displays a list of functions that are available for your transformations.

 **SQL Optimization button**

Opens the SQL APT Status report.

 **Check Expression button**

Verifies the validity of the expression.

 **Sample Data button**

Produces sample data for the expression.

 **Find button**

Allows you to search for a function or variable by name.

Calculator buttons

Insert numbers and operators.

Function Assist button

Allows you to specify parameters for the function through a dialog box when creating or editing a transformation.

The following operators are available:

| (single concatenation bar)

Concatenates two values, retaining any trailing blanks after the first one. For example, if FIRST_NAME and LAST_NAME were both in A15 format, the expression

```
FULL_NAME = FIRST_NAME | LAST_NAME
```

would produce a column like the following:

```
MICHAEL       SMITHSON
JANE          JONES
.
.
.
```

|| (double concatenation bar)

Concatenates two values, suppressing any trailing blanks in the first. For example, to construct the full name and insert a comma, the syntax

```
FULL_NAME = LAST_NAME || (', ' | FIRST_NAME)
```

would produce a column like the following:

```
SMITHSON, MICHAEL
JONES, JANE
.
.
.
```

The concatenation in the parentheses is done first (preserving the blank space after the comma), and the result is then concatenated to LAST_NAME, suppressing the trailing blanks of LAST_NAME.

IF

Establishes a conditional test.

THEN

Specifies the action to perform if the result of a conditional test is TRUE.

ELSE

Specifies the action to perform if the result of a conditional test is FALSE.

LT

Returns the value TRUE if the value on the left is less than the value on the right.

NOT

Returns the value TRUE if the operand is false.

LE

Returns the value TRUE if the value on the left is less than, or equal to the value on the right.

EQ

Returns the value TRUE if the value on the left is equal to the value on the right.

AND

Returns the value TRUE if both operands are true.

GT

Returns the value TRUE if the value on the left is greater than the value on the right.

GE

Returns the value TRUE if the value on the left is greater than, or equal to the value on the right.

NE

Returns the value TRUE if the value on the left is not equal to the value on the right.

OR

Returns the value TRUE if either operand is true.

**

Raises a value to the specified power.

( )

Adds parentheses.

''

Inserts two single quotation marks. Enter alphanumeric test values between these.

a->A

Converts selected text to uppercase.

A->a

Converts selected text to lowercase.

Date

Opens the Date Editor dialog box, which lets you use the current date or specify a date from the calendar.

Datetime

Opens the Date Editor dialog box, which lets you use the current date and time or specify a date and time from the calendar.

Order of Evaluation

The order of evaluation can affect the result of an expression in a transformation. Data Migrator evaluates numeric and Boolean operators from left to right in the following order:

- Operations enclosed in parentheses
- Exponentiation
- Multiplication and division
- Addition and subtraction
- EQ, NE, LE, LT, GE, GT, NOT

- AND
- OR

Changing the Order of Evaluation

The following expression gives an incorrect result because UNIT_SOLD is first multiplied by UNIT_COST, and then the result is subtracted from RETAIL_PRICE:

```
RETAIL_PRICE - UNIT_COST * UNIT_SOLD
```

You can change the order of evaluation and improve readability by enclosing expressions in parentheses. An expression in parentheses is evaluated first. The correct expression for the preceding example is:

```
(RETAIL_PRICE - UNIT_COST) * UNIT_SOLD
```

Joining Data Sources

If you select more than one data source for your data flow, you must use a Join object to connect them.

You need to specify the common column or columns that will link the tables, as well as the type of join.

You can perform an inner, left outer, right outer, full outer, or cross join.

- An inner join extracts those rows that appear in both tables. This is the default.
- A left outer join includes all of the rows from the table listed in the Left Source Columns grid, and the columns from the Right Source Columns grid that match one of these columns.
- A right outer join includes all of the rows from the table listed in the Right Source Columns grid, and the columns from the Left Source Columns grid that match one of these columns.
- A cross join is a Cartesian product of two tables. It consists of all possible pairs of rows from the two tables.
- A full outer join includes all of the rows that are extracted from both data sources.

Note: You can create a recursive join by selecting the same data source twice.

When you link data sources, Data Migrator identifies each table with T*n*, where *n* is the order in which the table was chosen. For example, the first source chosen would be identified as T1, the second as T2, and so on.

Data Migrator creates join syntax in ANSI 92 SQL. For example, when selecting the tables Product and Vendor, and columns VENDORNUM and VENDOR_NUM, Data Migrator generates the SQL syntax:

```
SELECT...FROM
Product t1 INNER JOIN VENDORS T2 on t1.vendornum=T2.vendor_num
```

Reference: Considerations for Join Performance

Data Migrator lets you create joins across similar data sources, different data sources, and across servers (also known as cross-platform joins).

In general, cross-platform joins, and joins without key fields, can create many intermediate files, depending on the number of tables joined. This issue may be less significant for some platforms. For example, the Server for z/OS can use hyperspaces and alternate sort utilities. If performance is a concern, you should carefully evaluate the impact of your joins.

To improve performance, try any of the following:

- ❑ Minimize the number of tables used in a join.
- ❑ Join tables using key columns and indexes, when possible.
- ❑ Make sure that the smaller of the two tables is selected first, and the larger one is selected second.

Procedure: How to Automatically Add Join Objects

The default settings of the DMC automatically create Join objects. If Join objects are not being created, you can change your settings so that Join objects are created automatically.

1. On the Home tab, in the Tools group, click *Options*.
2. Expand *Data Flow* and select *Data Flow Designer* from the tree.
3. Select the *Add Join Object if needed* check box in the On Add Source section.
4. Click *OK*.

Procedure: How to Join Data Sources

If the DMC is using the default settings, Join objects are created automatically when you drag additional data sources into the data flow workspace.

You can also add a join by right-clicking in the data flow workspace and selecting *Add Join*. In that case, you must connect the data source to the join by right-clicking it and dragging it to the Join object.

1. To complete the join, double-click the Join object, or right-click a Join object and select *Join Editor*.

   The Join Editor dialog box opens, listing the columns from the two data sources in the Left and Right Source Columns lists. For more information, see *Join Editor Dialog Box* on page 209.

   If the two data sources have a column with the same name and format, and that column is a key column in one of the sources, an inner join will automatically be created for those columns. This default join can be changed, if necessary.

2. If you want to join columns with different names, select one column from each list, and click the equal sign to create the join.

   The equality condition will appear in the Expression box under Join Conditions.

   Data Migrator supports both equi and non-equi (conditional) joins. In an equi-join, the condition is one of equality, while a non-equi join uses any other condition, such as greater than, less than, and so on.

   Note: You can delete the condition from the Expression box by selecting it and clicking the *Delete Condition* button.

   You can edit the condition by selecting it in the Expression box and clicking the *Edit Condition* button.

3. Select the type of join you want from the Join Type drop-down menu, by clicking the join diagram, or right-clicking anywhere in the diagram and selecting a join type from the menu.

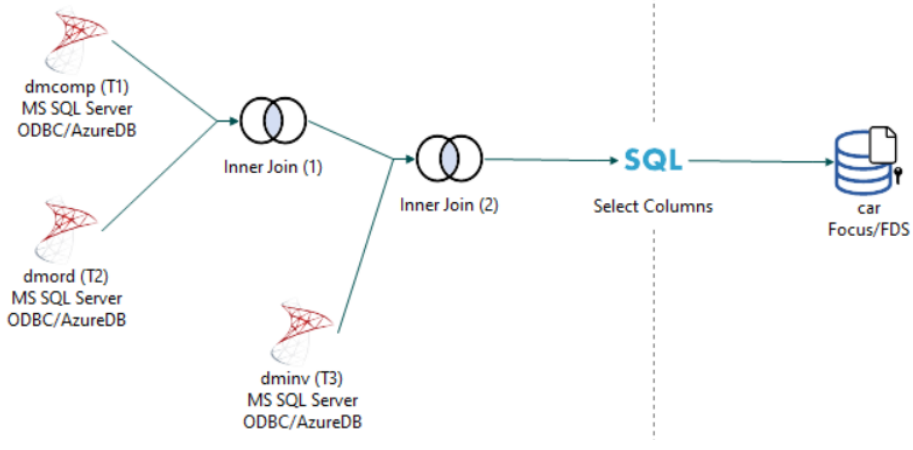
   - ❏ Click the area where the circles intersect for an inner join. This is the default.
   - ❏ Click the left circle for a left outer join.
   - ❏ Click outside the circles for a cross join or Cartesian product.
   - ❏ Click the right circle for a right outer join.
   - ❏ Click an inner circle a second time for a full outer join.

   The type of join you create will be graphically displayed in the workspace.

4. To add a calculation or SQL function to the condition, select it and click the *Edit Condition* button. When the Join Calculator opens, enter your calculation in the Conditions box by typing it, or by using the Columns/Variables and Functions tabs and the calculator buttons. For details, see *Join Calculator* on page 211. For information about using functions, see *Using an SQL Function* on page 230. For information about using variables, see *Using Variables in a Flow* on page 314.
5. To define a new condition, click the *Insert Condition* button, which will launch the Join Calculator. This is where you can create the condition.
6. Click the *Test Join* button to test the join.
7. Click *OK* to save the join.
8. Drag another synonym into the workspace, to the left of the SQL object.
9. Double-click the second Join object.
10. Select one column from each list, and click the equal sign to create the join.

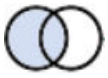
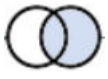
    The equality condition will appear in the Expression box under Join Conditions.
11. Click *OK*.

An example of a data flow with three joined sources is shown in the following image.



Reference: Join Objects in the Workspace

Join objects appear as follows in the workspace:

Object	Function
	Inner Join Indicates that two data sources are connected by an inner join and that only rows that appear in both tables used in the Join are extracted.
	Left Outer Join Indicates that two data sources are connected by a left outer join and that all rows are extracted from the left data source, as well as the columns from the right source that match.
	Right Outer Join Indicates that two data sources are connected by a right outer join and that all rows are extracted from the right data source, as well as the columns from the left source that match.
	Full Outer Join Indicates that two data sources are connected by a full outer join and that all rows are extracted from both data sources.
	Cross Join Indicates that two data sources are connected by a cross join, or Cartesian product of two tables. It consists of all possible pairs of rows between the two tables.

Procedure: How to Create a Join Condition With the Join Calculator

1. From the Join Editor dialog box, click the *Add new row* button, and then click the *Calculator* button.

   The Join Calculator opens.

2. Enter your calculation in the Expressions box by typing it, or by using the Columns/ Variables and Functions tabs and the calculator buttons. For information about using functions, see *Using an SQL Function* on page 230. For information on using variables, see *Using Variables in a Flow* on page 314.

3. Click *OK*.

Procedure: How to Edit a Join Condition With the Join Calculator

1. From the Join Editor dialog box, select the condition and click the *Edit Condition* button.

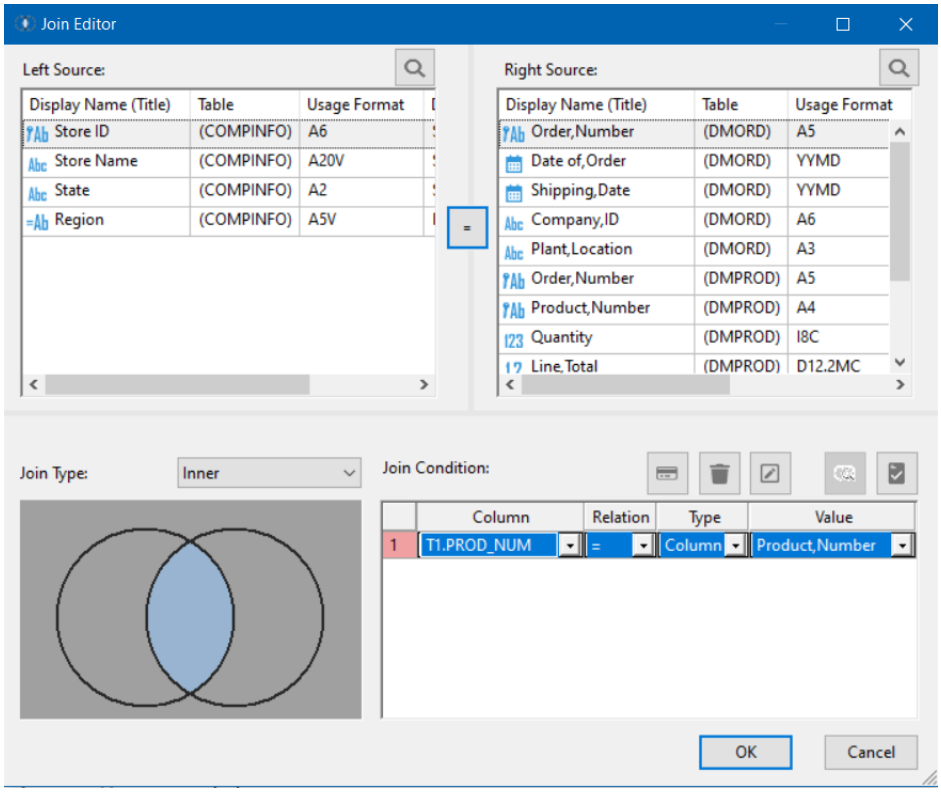
   The Join Calculator opens.

2. Edit your calculation in the Expressions box by typing it, or by using the Columns/Variables and Functions tabs and the calculator buttons.

3. Click *OK*.

Reference: Join Editor Dialog Box

To access the Join Editor dialog box from the workspace, double-click a Join object, or right-click a Join object and select *Join Editor*.



The Join Editor dialog box contains the following fields and options:

Left Source

Displays the available columns from the first data source.

Next to each column name, there is a symbol that indicates regular columns, key columns, index columns, or virtual columns.

Note: Clicking a column heading will sort the column grid on that heading.

Right Source

Displays the available columns from the second data source.

=

Establishes the relationship between the selected columns as equal.

 **Find**

Allows you to search for a column.

 **Add New Row**

Adds a row to create an additional condition.

 **Delete Condition**

Deletes the selected condition.

 **Edit condition**

Opens the Join Calculator to edit a condition.

 **Join Profiler**

Opens the Join Analysis window, where the join condition is analyzed to show the number of rows and distinct rows that would result from a left outer, inner, and right outer join.

 **Test join**

Tests the join conditions.

Note: If you have not selected columns, test join displays the keys columns from the tables in the join. If you have selected columns, it displays the columns selected. If neither table has keys, and no columns were selected, test join displays an empty answer set.

Join Conditions

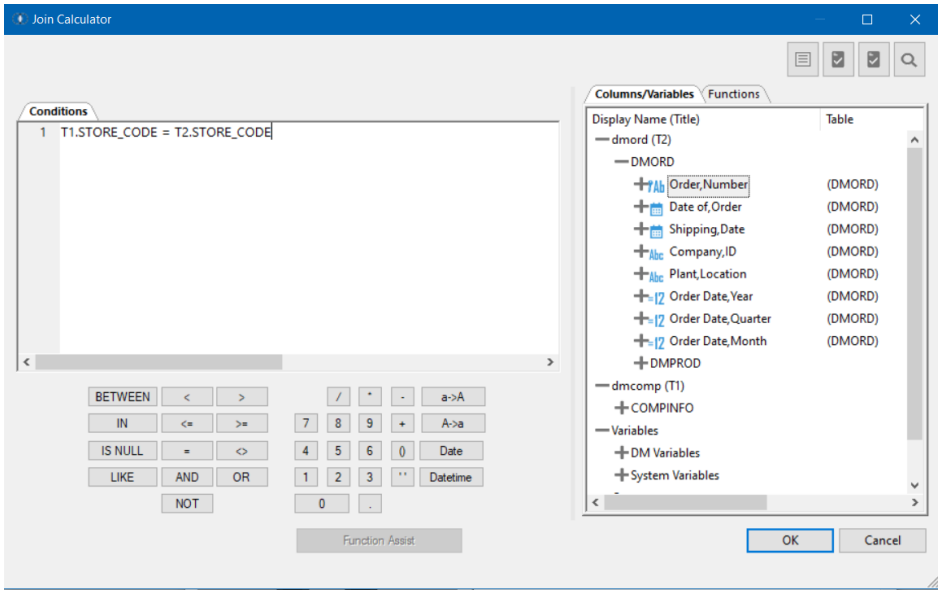
Lists the conditions by which the two data sources are connected.

Join Type

Enables you to select the join type from the drop-down menu or by clicking in the join diagram, which graphically represents the type of join being used.

Reference: Join Calculator

To access the Join Calculator from the Join Editor dialog box, select a row, and click the *Edit Condition* button.



The Join Calculator contains the following fields and options:

Conditions window

Displays the expression.

Columns/Variables tab

Displays available columns and variables.

Functions tab

Displays a list of SQL functions that are available for your transformations.

Function Assist button

Allows you to specify parameters for the function through a dialog box when creating or editing a transformation.

Calculator buttons

Insert numbers and operators.

The following operators are available in both the Join and Filter calculators:

BETWEEN

Specifies data will be returned if the value of the field falls between the values entered on either side of the word AND. For example:

```
RETAIL BETWEEN 20 AND 30
```

IN

For alphanumeric data, this option specifies data will be returned if the field value is a substring of the right operand.

IS NULL

Specifies that records with a null value for the specified column are returned.

LIKE

Returns data if the value of the field matches the character string on the right operand. In the character string, use the underscore character (_) to represent any single character, and the percent character (%) to represent any sequence of characters. For example, the following string returns the value TRUE if LAST_NAME contains the string SMITH anywhere inside it (for example, SMITH, SMITHSON, SILVERSMITH):

```
LAST_NAME LIKE '%SMITH%'
```

The following string returns the value TRUE if LAST_NAME is exactly five characters long and the first letter is S.

```
LAST_NAME LIKE 'S_ _ _ _'
```

<

Returns data if the value of the field is less than the right operand. For example, the following string restricts the query to those employees who earn less than $30,000:

```
SALARY < 30000
```

<=

Returns if the value of the field is less than, or equal to the right operand.

=

Specifies data if the field value equals the right operand.

AND

Is used in combination with the BETWEEN operator.

NOT

Is used in combination with other relational operators to negate them. For example, the following string returns all records where the value of LAST_NAME is not null:

```
LAST_NAME IS NOT NULL
```

>

Specifies data will be returned if the value of the field is greater than the right operand.

>=

Specifies data will be returned if the value of the field is greater than, or equal to the right operand.

<>

Specifies data will be returned if the left operand is not equal to the right operand.

OR

Inserts the operator OR.

( )

Adds parentheses.

''

Inserts two single quotation marks (‘). Enter alphanumeric test values between these.

a->A

Converts selected text to uppercase.

A->a

Converts selected text to lowercase.

Date

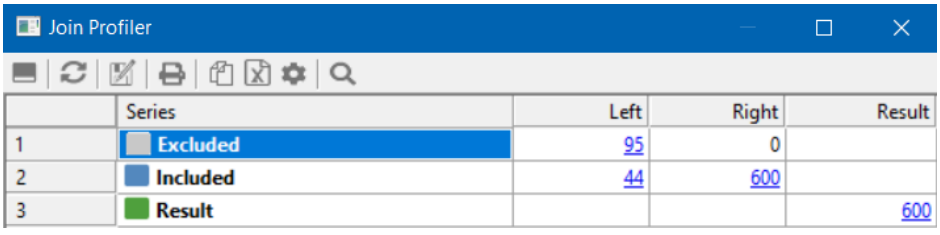
Opens the Date Editor window, which lets you use the current date or specify a date from the calendar.

Datetime

Opens the Date Editor window, which lets you use the current date and time or specify a date and time from the calendar.

Reference: Join Profiler Dialog Box

To access the Join Profiler dialog box from the Join Editor dialog box, click the *Join Profiler* button, and then click *Join Profiler*.



	Series	Left	Right	Result
1	Excluded	95	0	
2	Included	44	600	
3	Result			600

The first row shows, for the left and right data sources, how many rows would be excluded based on the join criteria.

The second row shows, for the left and right data sources, how many rows would be included based on the join criteria.

The third row shows how many rows would be returned based on the join criteria.

The Left and Right columns show the number of additional values and rows that would be returned for their respective join type.

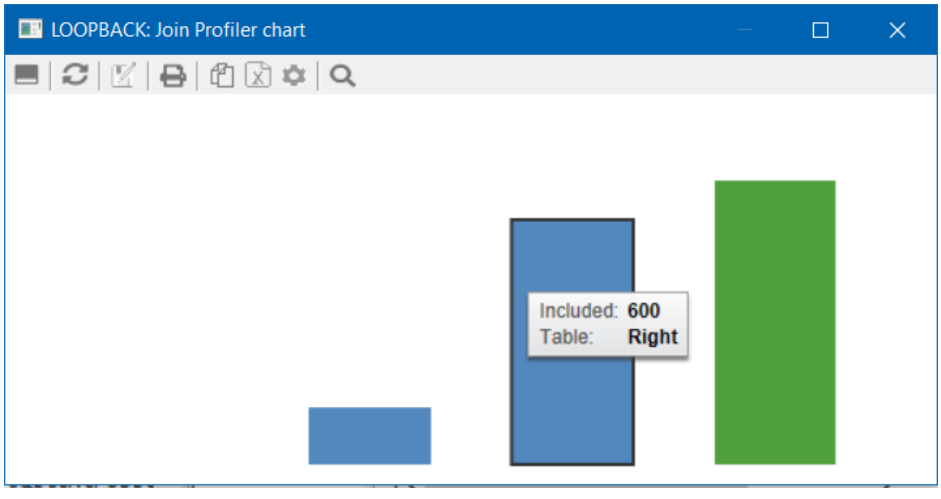
From the example shown in the above image, an inner join would exclude 95 rows from the left data source.

Note that all of the numbers in each row or column are also hotlinks. For example, if you double-click *44*, a Test Results window opens, as shown in the image below. This window shows the 43 values for Store ID that were in the left source, but that were not found in the right source.

	Store ID	Count
1	1003DC	3
2	1003GA	2
3	1003IL	4
4	1003LA	2
5	1003MA	3
6	1003MD	2
7	1003MO	1

Reference: Join Profiling Chart

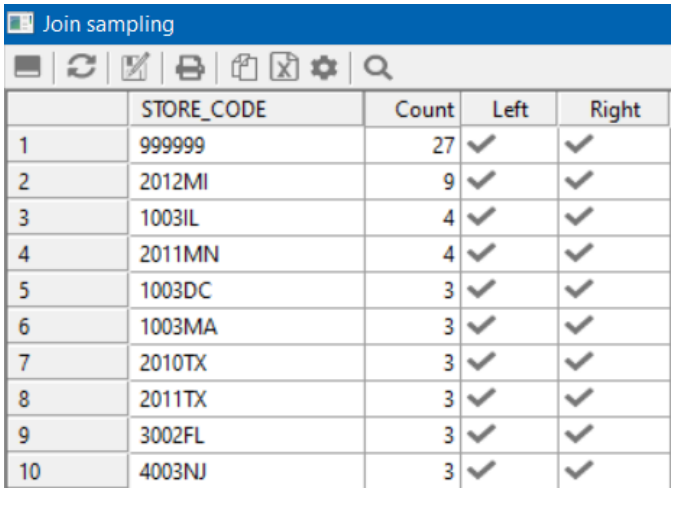
To access the Join Profiler Chart from the Join Editor dialog box, click the *Join Profiler* button, and then click *Join profiler chart*.



The chart shows a vertical bar chart that represents the number of rows that would be returned from each table when a Left Outer, Right Outer, or Inner Join is run.

Reference: Join Sampling

To access the Join Sampling report from the Join Editor dialog box, click the *Join Profiler* button, and then click *Join sampling*.

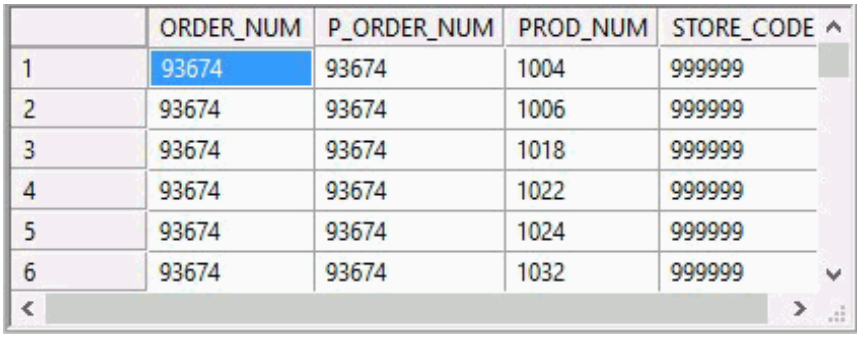

Join sampling

	STORE_CODE	Count	Left	Right
1	999999	27	✓	✓
2	2012MI	9	✓	✓
3	1003IL	4	✓	✓
4	2011MN	4	✓	✓
5	1003DC	3	✓	✓
6	1003MA	3	✓	✓
7	2010TX	3	✓	✓
8	2011TX	3	✓	✓
9	3002FL	3	✓	✓
10	4003NJ	3	✓	✓

The report shows, for each value of the join fields, the number of rows returned and whether each row was returned from the left and right sources.

Reference: Test Individual Join Window

To access the Test Individual Join window from the Join Editor window, select a join and click the *Test Join* button. Sample data from the join will be shown, as shown in the following image.



	ORDER_NUM	P_ORDER_NUM	PROD_NUM	STORE_CODE
1	93674	93674	1004	999999
2	93674	93674	1006	999999
3	93674	93674	1018	999999
4	93674	93674	1022	999999
5	93674	93674	1024	999999
6	93674	93674	1032	999999

Reference: Join Profiler Buttons

Each report returned by the Join Profiler has the following options.

Run Options

Opens a dialog box for entering or selecting the following run options:

- **Maximum number of rows for test reports.** The default value is 50.
- **Maximum number of columns for test reports.** The default value is 999999.
- **Test reports default format.** The default value is *Default*. Other selections are HTML, HTML - plain text, Active Report, PDF, or Unformatted.
- **Omit missing (null) values in Data Profiling charts.** This box is deselected, by default. To omit null values, check this box.
- **Defaults for New Flows:**

  Stop processing if 0 rows selected. This option is selected, by default.

  Stop after DBMS errors: The default value is 10000000.

Refresh

Refreshes the current document.



Save As

Opens the Save As dialog box to save the active document, with a new name.



Print

Prints the active document.



Copy as Text

Copies selected data as text. You can paste it into any type of file that accepts text input.



Export to Excel

Exports selected data to Excel.



Options

Enable you to choose the columns to display or restore the display to the defaults.



Find

Opens the Find dialog box. You can enter text to search for in the active document. You can choose to search backward or forward (the default), match case, and search for whole words only.

Reference: Join Object Options

After a join has been added to the workspace, the following options are available in the shortcut menu:

Expand/Collapse

Opens or closes a window that allows you to view the join expression and sample data.

Join Editor

Opens the Join Editor which allows you to create or edit the join conditions.

Add

Opens a submenu with the following options. Note that these options are only available when the Join object does not already have two inputs.

- **Source.** Opens the Select Synonym panel to add a new synonym to the data flow connected to this Join object. When adding the second synonym the Select Synonym opens with Show synonyms with matching keys filter already enabled.
- **Join.** Adds a new Join object as input to this Join object.

- **Select.** Adds a new Select object as input to this object.
- **Union.** Adds a new Union object as input to this object.

Delete

Deletes the selected Join object.

Properties

Opens the Properties panel for the selected Join object.

Error Info

Displays a dialog box with an error message. This option is only available if there is an error. For more information, click *Details*.

Adding a SELECT to a Data Flow

Adding a SELECT to a data flow lets you add a subselect, a select within a select. A subselect, also called a subquery, allows you to use the results of one select as part of another select. While subselects can be used in different places in the SQL language, adding a select to a data flow lets you use a select statement as a source, for example, to join to an aggregated or filtered selection from a table. By using a subselect, the need for intermediate staging tables can be avoided.

Procedure: How to Add a SELECT to a Data Flow

From the data flow workspace:

1. Right-click to the left of the SQL object and click *Add Select*.

   or

   Right-click a SQL, Join, or Union object, point to *Add*, and then click *Select*.

2. A new SQL object appears in the data flow. Right-click the object, point to *Add*, and then click another object type.
3. Continue to build the data flow.

Adding a UNION to a Data Flow

Adding a union to a data flow lets you combine the results of two select statements into a single result set.

There are some restrictions on the select statements that can be used:

- The two select statements must return the same number of columns.

- The data type of each column in the first table must be the same as the data type of the corresponding column in the second table.
- Neither of the select statements can include a sort (ORDER BY). However, the combined results may be sorted.

Procedure: How to Add a UNION to a Data Flow

From the data flow workspace:

1. Right-click to the left of a SQL object and click *Add Union*.

   or

   Right-click a SQL, Join, or Union object, point to *Add*, and then click *Union*.

   or

   Drag the *Union* object from the Insert group of the Flow tab, into the workspace.
2. A new Union object appears in the data flow. Right-click the object, select *Add*, and then click *Union*.
3. Continue by adding objects to the select as needed.
4. Right-click the Union object, point to *Add*, and then click *Select* (each union requires two select statements).
5. Continue by adding objects to the second select as needed.

   Note: You can test the union by double-clicking the object and clicking the *Test SQL* tab to see a sample of the data.
6. Right-click the Union object and click *Properties*.
7. The Properties pane opens. Select the Union type. See *Union Types* on page 220 for details.

Reference: Union Types

The Union object can be used to combine answer sets in different ways, depending on the type selected in the Properties pane. The different union types are:

Object	Function
	Union Returns a result that contains all rows from both sets with duplicate rows removed.
	Union All Returns a result that contains all rows from both sets, including duplicates.
	Except Returns a result set that contains all rows from the first set, excluding common rows from the second set.
	Except All Returns a result set that contains all rows from the first set, excluding common rows from the second set, including duplicates.
	Intersect Returns a result set that contains the common rows from two result sets.
	Intersect All Returns a result set that contains the common rows from two result sets, including duplicates.

Selecting Columns

The columns you select define the input to your data target. If you are using the DMC to format a new data target, the order of the columns will reflect the order that the columns are specified in the SQL command.

The Server, which Data Migrator uses to process its extraction logic, can have up to approximately 400 columns in a single data flow. If you intend to move more than 400 columns, several data flows may be linked together by creating a process flow. For information on process flows, see *Designing a Process Flow* on page 369.

Procedure: How to Automatically Select All Columns

The default settings of the DMC require you to manually select the columns you want to extract. To change your settings so that all columns are selected automatically:

1. On the Home tab, in the Tools group, click *Options*.
2. Expand *Data Flow* and select *Data Flow Designer* from the tree.
3. Select the *Automatically select all columns* check box in the On Add Source section.
4. Click *OK*.

Procedure: How to Select Columns

1. Right-click the SQL object in the data flow workspace and click *Column Selection*.

   The Column Selection dialog box opens.
2. Double-click the columns you want to select from the Available Columns grid on the left.

   or

   Select one or more columns and click the right arrow.

   Tip: To select all columns, click anywhere in the column list, and press Ctrl + A.

   The columns appear in the Selected Columns grid on the right.
3. You can change the order of the columns by selecting them and clicking the up and down arrows.
4. Optionally, you can select the *Distinct* check box to remove duplicate rows from the results of your query.
5. You can rename a selected column by entering a name in the Sql Alias field.
6. To remove a column from the Selected Columns grid, select it and click the left arrow, or click the *Delete* button.
7. Click *OK*.

Procedure: How to Edit Selected Columns

1. Right-click the SQL object in the data flow workspace and click *Column Selection*.

   The Column Selection dialog box opens.

2. Select the column you want to edit.
3. Click the *Edit column* button, or double-click the number next to the column in the Selected Columns grid.

   The SQL Calculator opens.
4. Using the Columns/Variables and Functions tabs and the calculator buttons, edit the expression for the column. For information about using functions, see *Using an SQL Function* on page 230. For information about using variables, see *Using Variables in a Flow* on page 314.
5. Click *OK*.

For information about creating a calculated column in the Selected Columns grid, see *Creating an SQL Calculation* on page 226.

Procedure: How to Add or Change Column Descriptions From the Column Selection Dialog Box

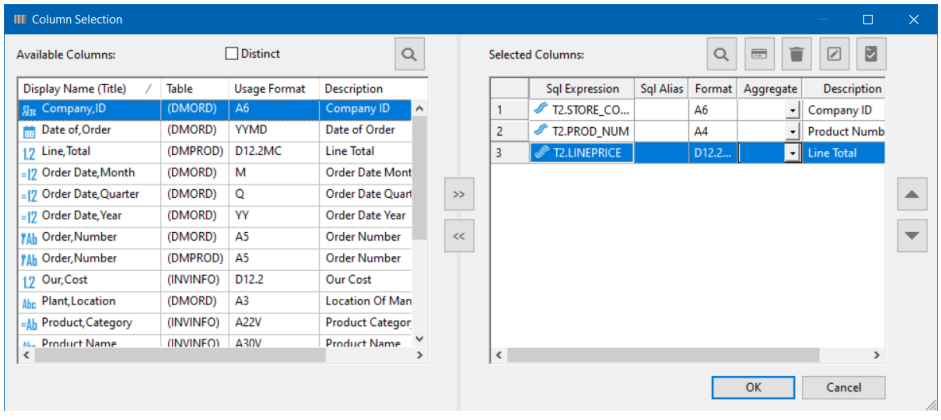
If a column in a source synonym has a description, that description will be carried over with the column into any new target synonyms. You can also use the Column Selection dialog box to add descriptions for columns that do not have them, or to change the descriptions for columns that do have them. These descriptions will also be carried over into any new target synonyms in the current flow.

1. From the data flow workspace, double-click the *SQL object*, or right-click it and click *Column Selection*.

   The Column Selection dialog box opens.
2. Double-click the *Description* field for the column in the *Selected Columns* grid.
3. Type the description.
4. Click *OK*.

Reference: Column Selection Dialog Box

To access the Column Selection dialog box from the data flow workspace, double-click the *SQL* object, or right-click it and click *Column Selection*.



The Column Selection dialog box contains the following fields and options by default:

Distinct

Removes duplicate rows from the results of your query.

Available Columns

Lists the columns in each data source.

Right arrow

Adds columns to the Selected Columns grid.

Left arrow

Deletes columns from the Selected Columns grid.

Selected Columns

Lists the columns to be extracted from the data source(s).

Sql Expression

Is the column name, or an expression created in the SQL Calculator.

Sql Alias

Is an alternate name for referencing the column. If you add an alias and expression, the expression shows up where the column name would normally be.

Format

Is the format of the column.

Aggregate

Is the aggregation applied to the column. The available options are: Group By, Group By Not Selected, Sum, Min, Max, Count, Avg, Sum Distinct, Count Distinct, .

Description

Is the description of the column.

Nulls

Indicates whether the column can contain NULL values.

Find

Allows you to search for a column.

 **Insert columns**

Opens the SQL Calculator to create a column.

 **Delete columns**

Deletes the selected column.

 **Edit condition**

Opens the SQL Calculator to edit a column.

 **Test SQL Statement**

Tests the SQL statement and displays the result.

Note: The test button will be grayed out if you have not selected any columns.

Up and down arrows

Moves the selected column up or down within the Selected Columns grid.

If you right-click anywhere on the header bar, you get the following options:

Find

Allows you to search for a column.

Best Fit

Resizes the column to fit its data.

Customize

Allows you to select what information is displayed for each column.

If you right-click any column, you get the following options:

Edit

Opens the SQL Calculator where you can edit the properties of a column.

Delete

Deletes the selected column.

Aggregating Columns

You can aggregate your columns in the Column Selection dialog box.

Procedure: How to Aggregate a Column

In the Column Selection dialog box:

1. Select the column or columns you want to aggregate. To select more than one column, hold the Shift or Ctrl keys while selecting columns.
2. Click the down arrow in the Aggregate field to display the drop-down menu containing the types of aggregation, and select an aggregation. Your options are:
   - **Group By.** Groups the data by each distinct value in the column.
   - **Group By Not Selected.** Groups the data on (or by) the column that you do not want to appear in the results.
   - **Sum.** Adds the values within the column.
   - **Min.** Returns the minimum value found in the column.

- **Max.** Returns the maximum value found in the column.
- **Count.** Returns a count of all non-null column values.
- **Avg.** Returns the average value for the column.
- **Sum Distinct.** Adds the distinct values within the column, excluding duplicates.
- **Count Distinct.** Returns a count of all distinct, non-null column values, excluding duplicates.
- **Avg Distinct.** Returns the average value for the column, excluding duplicates.

Note: If you apply an aggregate function to one of your selected columns, the rest of the selected columns must, in most cases, have aggregation functions applied, or be GroupBy columns. After an aggregation is applied to a column, all columns without an aggregation turn red until an aggregation is applied to them as well.

Creating an SQL Calculation

You can create an SQL calculation for your data flow that:

- Renames a column.
- Performs a calculation. For details, see *How to Create a Calculation* on page 226.
- Designates a constant value for a column.
- Uses a variable to calculate a column. For details, see *How to Create a Column That Uses a Variable* on page 227.

Procedure: How to Create a Calculation

1. In the Column Selection dialog box, click the *Insert Columns* button .

   The SQL Calculator opens.

2. Enter the calculation you want to perform using the Columns/Variables and Functions tabs, and the calculator buttons. For details, see *SQL Calculator* on page 228. For information about using functions, see *Using an SQL Function* on page 230. For information about using variables, see *Using Variables in a Flow* on page 314.
3. In the Alias box, enter an alias for the column.
4. Click *OK*.

   The calculation appears in the SQL Expressions field.

   Note: When you create expressions, keep in mind their order of evaluation. For details, see *Order of Evaluation* on page 203.

5. Optionally, select an aggregation for the column in the *Aggregate* field. For details, see *Aggregating Columns* on page 225.

   Note:

   - You can type a calculation directly into the Selected Columns grid by double-clicking the Sql Expressions field.
   - You can type an alias directly into the Selected Columns grid by double-clicking the Sql Alias field.

Procedure: How to Create a Column That Uses a Variable

From the Column Selection dialog box:

1. Click the *Insert Columns* button.

   The SQL Calculator opens.

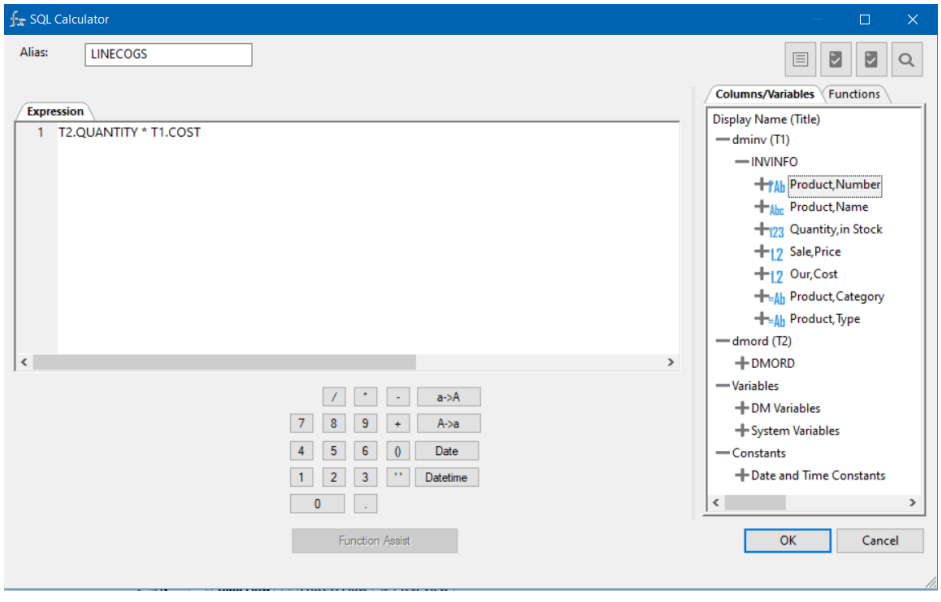
2. Type an alias for the virtual column.
3. Double-click the variable you want assigned to the column from the Columns/Variables tab. You can also type in character-valued or date-valued variables, which must be enclosed in single quotation marks (‘).
4. Click *OK*.

The variable and alias appear in the Selected Columns grid.

For more information about using variables, see *Using Variables in a Flow* on page 314.

Reference: SQL Calculator

To access the SQL Calculator from the Column Selection dialog box, click the *Insert Columns* button, or select a column and click the *Edit Column* button.



The SQL Calculator contains the following fields and options:

Alias

Is an alias for the column.

Expression window

Displays the expression.

SQL Optimization button

Opens the Filter SQL Optimization Report, which allows you to specify settings for, and run, the report.

Check Expression button

Verifies the validity of the expression.

Test SQL Statement

Tests the SQL statement and displays the result.

Find button

Allows you to search for a function or variable by name.

Columns/Variables tab

Displays available columns and variables in a hierarchical tree or in a grid.

Functions tab

Displays a list of SQL functions that are available for your transformations.

Function Assist button

Allows you to specify parameters for the function through a dialog box when creating or editing a transformation.

Calculator buttons

Insert numbers and operators.

( )

Adds parentheses.

''

Inserts two single quotation marks (‘). Enter alphanumeric test values between these.

a->A

Converts selected text to uppercase.

A->a

Converts selected text to lowercase.

Date

Opens the Date Editor dialog box, which lets you use the current date or specify a date from the calendar.

Datetime

Opens the Date Editor dialog box, which lets you use the current date and time or specify a date and time from the calendar.

Reference: SQL Operators

The following table describes the result of an operator on an expression, where *expr* is an expression.

Operator	Result
expr + *expr*	Returns the sum of two expressions.
expr - *expr*	Returns the difference of two expressions.
- *expr*	Returns the negated value of an expression.
expr * *expr*	Returns the product of two expressions.
expr / *expr*	Returns the dividend of two expressions.
expr || *expr*	Returns the concatenation of two character strings.

Using an SQL Function

You can use an SQL function in the SQL Calculator to create a complex expression. In addition, most ANSI 92 SQL functions can be used in a SELECT command, depending on the data source type and platform.

Procedure: How to Use an SQL Function

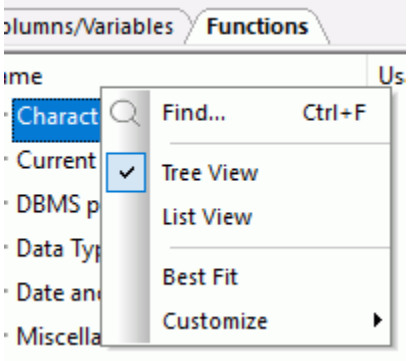
1. In the SQL Calculator, click the *Functions* tab.
2. Double-click the folder for the type of function you want to use.
3. Double-click the function you want to use.

   The Function Assist dialog box opens.
4. Use the Function Assist dialog box to set the parameters for the function and click *OK*.
5. The function appears in the Expressions dialog box. Click *OK*.
6. The function is added to the Selected Columns grid of the Column Selection dialog box.

For more information about the Function Assist dialog box, see *How to Use Function Assist* on page 342.

Reference: SQL Functions

You can view the functions in the SQL Calculator Functions tab by category (Tree View) or alphabetically (List View). Right-click the *Name* header bar to select between the two views, as shown in the following image. For more information about SQL functions, see the SQL function chapters in the *TIBCO® Data Migrator Functions Reference* manual.



SQL Character Functions	Description
`CHAR_LENGTH(arg)`	Returns the length of *arg* in a character string.
`CONCAT(arg1, arg2)`	Concatenates the values of two arguments.
`EDIT(arg, mask)`	Edits a numeric or character value according to a format specified by a mask.
`GET_TOKEN(string, delimiter_string, occurrence)`	Extracts a token from an input string.
`INITCAP(string)`	Capitalizes each word in a string, where a word is delimited by white space or characters that are not alphanumeric.
`LCASE(arg)`	Converts a character string value to lowercase. Identical to LOWER and LOWERCASE.
`string LIKE mask [ESCAPE escape]`	Select records that conform to a pattern.

SQL Character Functions	Description
LPAD(*string, out_length, pad_character*)	Left pads a character string with the pad character to the length specified in out_length.
LTRIM(*arg*)	Removes leading spaces from a character string.
PATTERNS(*string*)	Returns a string that represents the pattern of the input string.
POSITION(*substring* IN *arg*)	Returns the location of *substring* in *arg*, or returns a zero if *substring* is not found.
REPLACE(*string, pattern,* [*replacement*])	Replaces the pattern in the pattern in the string with the replacement (if supplied) or removed.
REVERSE(*string*)	Reverses a character string.
RIGHT(*source_string, substring_limit*)	Returns the right portion of an input string.
RTRIM(*arg*)	Removes trailing spaces from a character string.
string RLIKE regular_expression	Matches a string to a Regular Expression.
RPAD(*string, out_length, pad_character*)	Right pads a character string with the pad character to the length specified in out_length.
RTRIM(*arg*)	Removes trailing spaces from a character string.
SPLIT(*part, string*)	Returns a part from a string.
SUBSTR(*arg* FROM *start-pos*[FOR *length*])	Returns a string from *arg* starting at *start-pos* to the end of the string, or if specified *length* characters.

SQL Character Functions	Description
SUBSTR(*arg, start_pos*[, *length*])	Returns a string from *arg* starting at *start-pos* to the end of the string, or if specified *length* characters.
TOKEN(*string, delimiter, number*)	Extracts a token from a string.
TRIM(*arg*) TRIM(*trim_char* FROM *arg*) TRIM(*trim_where*[*trim_char*] FROM *arg*)	Removes leading and/or trailing spaces and other characters from a character string. The character to be removed may be specified as *trim-char*. If it is not, the space character is assumed. Whether to remove leading and/or trailing characters may be specified as *trim-where*. If not, both leading and trailing spaces are removed.
UCASE(*arg*)	Converts a character string value to uppercase. Identical to UPPER and UPPERCASE.

SQL Current Date and Time Functions	Description
CURRENT_DATE	Returns the date in YYMD format.
CURRENT_TIME[(*precision*)]	Returns the current time of the operating system in the form, HHMMSS.
CURRENT_TIMESTAMP [(*precision*)]	Returns the current timestamp of the operating system (date and time) in the form, YYYYMMDDHHMMSS.
CURRENT_TIMEZONE()	Returns the current time zone.

SQL Data Type Conversion Functions	Description
`CAST(expression AS data_type[(length)])`	Converts the value of its argument to a specified data type. *Length* is an optional integer that specifies the length of the target data type.
`CHAR(arg[,date_format])`	Converts a number to a character string, or data value to a string in a specified format.
`DATE(arg)`	Returns a date, time, or timestamp value that is computed from *arg*.
`DECIMAL(arg, [length [dec_places]])`	Converts a number into a fixed-length decimal where *dec-places* is the number of decimal places in the result.
`DIGITS(arg)`	Converts a numeric expression to a fixed length character string. Db2 only.
`DT_FORMAT(arg, date_format)`	Converts a date/timestamp expression to a character string.
`FLOAT(arg)`	Converts *arg* to a floating point value.
`FOCDATE(arg, format)`	Converts an alphanumeric expression to a date.
`INT(arg)`	Converts a number to an integer. Identical to INTEGER.
`OLDDATE(arg, format)`	Converts a date expression to an alphanumeric value.
`PHONETIC(string)`	Returns the phonetic key of the string.
`SMALLINT(arg)`	Converts a number to a small integer.
`TIME(arg)`	Converts the argument to a time.
`TIMESTAMP(arg)`	Converts the argument to a timestamp.

SQL Data Type Conversion Functions	Description
VARGRAPHIC(*arg*)	Converts the argument to a vargraphic.

SQL Date and Time Functions	Description
DAY(*arg*)	Returns a number that contains the day of the month from a date or timestamp.
DAYS(*arg*)	Returns the number of days since December 31, 1900.
DAY_OF_YEAR(*arg*)	Returns an integer from 1 to 366 that represents the day of the year for a date or date-time expression.
DTDIFF(*end_date, start_date, component*)	Returns the number of component boundaries between two dates.
DTIME	Returns the value of all of the time components up to and including the requested component. The remaining time components in the value are set to zero.
DTRUNC(*date_or_timestamp, date_period*)	Returns the date, which is the first day of the date period (YEAR, QUARTER, MONTH, DAY, etc.), encompassing the given date or timestamp.
datetime + INTERVAL *increment* component	Returns a new date or date-time value by adding an interval to a date or date-time expression.
EXTRACT(*field* FROM *arg*)	Extracts a numeric value from a date or time value *arg*, where *field* is either a year, month, day, hour, minute, or second.
HOUR(*arg*)	Returns the hour field from a time or timestamp value.

SQL Date and Time Functions	Description
MICROSECOND(*arg*)	Returns the number of microseconds from a time or timestamp value.
MILLISECOND(*arg*)	Returns the number of milliseconds from a time or timestamp value.
MINUTE(*arg*)	Returns the number of minutes from a time or timestamp value.
MONTH(*arg*)	Returns the number of the month in which *arg* falls.
QUARTER(*arg*)	Returns an integer value representing the quarter of the year for a date or date-time expression.
SECOND(*arg*)	Returns the second field from a time or timestamp value.
WEEKDAY(*arg*)	Returns an integer value between 1 and 7 representing the day of the week for a date or date-time expression.
YEAR(*arg*)	Returns the year in which *arg* falls.

SQL DBMS Pass-Through Function	Description
DB_EXPR(*sql_expression*)	Inserts the SQL expression as-is into the native SQL generated for an SQL language request.

SQL Miscellaneous Functions	Description
COUNTBY(*arg*)	Produces a column whose values are incremented row-by-row by a specified amount.
CURRENT_EDASQLVERSION()	Retrieves the current SQL parser version.

SQL Miscellaneous Functions	Description
GREATEST(*arg1*,*arg2*[,...*arn*])	Finds the maximum value.
HEX(*arg*)	Converts the input value to hexadecimal. **Note:** This function is available only for Db2, Ingres, and Informix.
IF(*test, val1, val2*)	Tests a condition and returns a value based on whether the condition is true or false.
LEAST(*arg1*,*arg2*[,...*arn*])	Finds the minimum value.
LENGTH(*arg*)	Returns the number of bytes of storage used by *arg*.
USER()	Returns the user ID running the query.

SQL Numeric Functions	Description
ABS(*arg*)	Returns the absolute value of a number.
CEIL(*number*)	Returns the smallest integer that is greater than or equal to the number.
EXP(*arg*)	Returns e raised to the power of the argument.
FLOOR(*number*)	Returns the largest integer that is less than or equal to the number.
LOG(*arg*)	Returns the natural logarithm of the input value.
MOD(*dividend, divisor*)	Calculates the remainder from a division as an integer.
POWER (*arg1, arg2*)	Returns the first number raised to the power of the second.

SQL Numeric Functions	Description
`RAND(seed)`	Returns a stream of random numbers uniformly distributed between 0 and 1.
`SIGN(arg)`	Returns a value (1 or -1) identifying the sign of the number.
`SQRT(arg)`	Returns the square root of the input value.

SQL Operators	Description

`CASE test-expr`
`  WHEN value-expr-1 THEN result-expr-1`
`  . . .`
`  WHEN value-expr-n THEN result-expr-n`
`  [ ELSE else-expr ]`
`END`
	Allows a value to be computed depending on the values of expressions.

`CASE`
`  WHEN cond-1 THEN result-expr-1`
`  . . .`
`  WHEN cond-n THEN result-expr-n`
`  [ ELSE else-expr ]`
`END`
	Allows a value to be computed depending on the truth or falsity of conditions.

`COALESCE(arg1, arg2, [ ... argn ])`	Can take two or more arguments. The first argument that is not NULL, is returned. If all arguments are NULL, NULL is returned.
`EXISTS(SELECT * FROM LOOKUP_MFD SQ [WHERE condition] )`	Tests if subquery returns one or more rows.
`NULLIF(arg1, arg2)`	Returns NULL if its two arguments are equal. Otherwise, the first argument is returned.
`test_exp1 IN (exp1, exp2)`	Determines whether a specified value matches any value in a list.

SQL Operators	Description
`(SELECT lookup_result from LOOKUP_MFD [WHERE condition])`	Used in a select list to return a column value.
`(SELECT * FROM LOOKUP_MFD SQ [WHERE condition] )`	Returns result set based on subquery.
`(test_exp1, test_exp2) IN ( SELECT exp1, exp2 FROM mfd WHERE condition)`	Determines whether a specified value matches any value in a subquery.

SQL Aggregation Functions	Description
`ASQ(field)`	Computes the average sum of squares of the field.
`AVG(field)`	Computes the average of the field.
`AVG(DISTINCT field)`	Computes the average of the distinct values of the field.
`COUNT(field)`	Counts the number of occurrences of the field.
`COUNT(DISTINCT field)`	Counts the number of distinct values in the field.
`MAX(field)`	Computes the maximum value of the field.
`MEDIAN(field)`	Computes the median of the field values.
`MIN(field)`	Computes the minimum value of the field.
`MODE(field)`	Computes the mode of the field values.
`SUM(field)`	Computes the sum of the field values.
`SUM(DISTINCT field)`	Computes the sum of the distinct values in the field.

SQL Analytic Functions	Description
`AVG(exp) OVER([PARTITION BY part] ORDER BY ord [window])`	Calculates the average over a group of lines determined by a partition and window.
`COUNT(exp) OVER([PARTITION BY part] ORDER BY ord [window])`	Calculates the count over a group of lines determined by a partition and window.
`DENSE_RANK(exp) OVER([PARTITION BY part] ORDER BY ord])`	Calculates the dense rank over a group of lines determined by a partition.
`FIRST_VALUE(exp) OVER([PARTITION BY part] ORDER BY ord [window])`	Calculates the first value of an ordered set of lines determined by a partition and window.
`LAG(exp, offset, default) OVER([PARTITION BY part] ORDER BY ord])`	Retrieves values from a previous row over a group of lines determined by a partition.
`LAST_VALUE(exp) OVER([PARTITION BY part] ORDER BY ord [window])`	Calculates the last value of an ordered set of lines determined by a partition and window.
`LEAD(exp, offset, default) OVER([PARTITION BY part] ORDER BY ord])`	Retrieves values from a future row over a group of lines determined by a partition.
`MAX(exp) OVER([PARTITION BY part])`	Calculates the maximum over a group of lines determined by a partition and window.
`MEDIAN(exp) OVER([PARTITION BY part] ORDER BY ord [window])`	Calculates the median of a group of lines determined by a partition.
`MIN(exp) OVER([PARTITION BY part])`	Calculates the minimum over a group of lines determined by a partition and window.
`MODE(exp) OVER([PARTITION BY part] ORDER BY ord [window])`	Calculates the mode of a group of lines determined by a partition.
`PERCENT_RANK(exp) OVER([PARTITION BY part] ORDER BY ord])`	Calculates the percent rank over a group of lines determined by a partition.

SQL Analytic Functions	Description
`RANK(exp) OVER([PARTITION BY part] ORDER BY ord])`	Calculates the rank over a group of lines determined by a partition.
`STDDEV_POP(exp) OVER([PARTITION BY part])`	Calculates the population standard deviation over a group of lines determined by a partition and window.
`STDDEV_SAMP(exp) OVER([PARTITION BY part])`	Calculates the sample standard deviation over a group of lines determined by a partition and window.
`SUM(exp) OVER([PARTITION BY part] ORDER BY ord [window])`	Calculates the sum over a group of lines determined by a partition and window.

SQL Statistical Functions	Description
`CORRELATION(input_field1, input_field2)`	Calculates the correlation between two fields.
`STDDEV_POP(field)`	Calculates population standard deviation for the field.
`STDDEV_SAMP(field)`	Calculates sample standard deviation for the field.

SQL Trigonometric Functions	Description
`ACOS(number)`	Calculates the angle in radians whose cosine is the specified value.
`ASIN(number)`	Calculates the angle in radians whose sine is the specified value.
`ATAN(number)`	Calculates the angle in radians whose tangent is the specified value.

SQL Trigonometric Functions	Description
ATAN2(*x*,*y*)	Calculates the angle in radians between the positive x-axis and the vector from the origin to the specified point.
COS(*angle*)	Calculates the cosine of the angle given in radians.
COT(*angle*)	Calculates the cotangent of the angle given in radians.
DEGREES(*angle*)	Converts the angle given in radians to degrees.
PI()	Returns the number pi.
RADIANS(*angle*)	Converts the angle given in degrees to radians.
SIN(*angle*)	Calculates the sine of the angle given in radians.
TAN(*angle*)	Calculates the tangent of the angle given in radians.

Testing the SQL Statement

Once you have created your SQL calculation, you should test it to see if you are selecting the data you need. The SQL object enables you to retrieve a limited set of rows and review them as part of the process of designing your data flow.

Note: You can set the number of rows retrieved from the Options dialog box in the Run Options pane. The default is 50 rows.

You can also edit the SQL code.

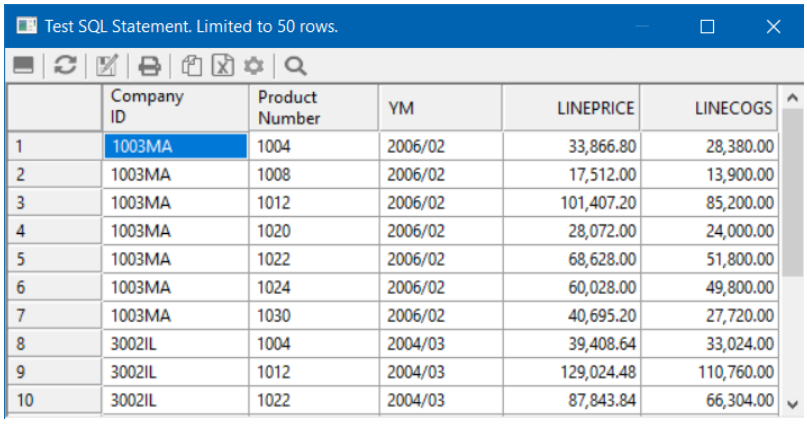
Tip: You can move SQL SELECT statements between applications by copying and pasting.

Procedure: How to Test the SQL

In the data flow workspace:

Double-click the *SQL* object and click the *Test SQL Statement* button in the upper-right corner of the Column Selection dialog box.

The Test SQL Statement Columns dialog box opens, which shows the sample data for the SQL object, as shown in the following image.



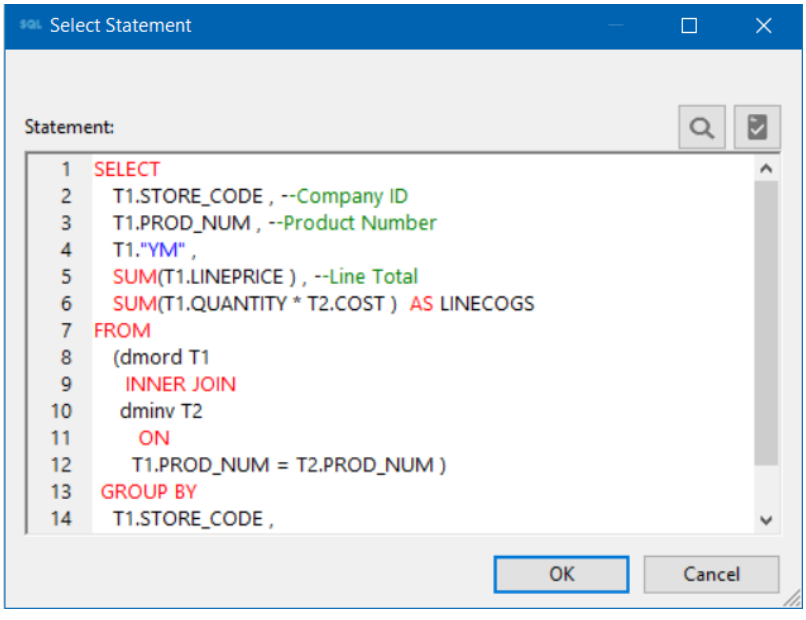
	Company ID	Product Number	YM	LINEPRICE	LINECOGS
1	1003MA	1004	2006/02	33,866.80	28,380.00
2	1003MA	1008	2006/02	17,512.00	13,900.00
3	1003MA	1012	2006/02	101,407.20	85,200.00
4	1003MA	1020	2006/02	28,072.00	24,000.00
5	1003MA	1022	2006/02	68,628.00	51,800.00
6	1003MA	1024	2006/02	60,028.00	49,800.00
7	1003MA	1030	2006/02	40,695.20	27,720.00
8	3002IL	1004	2004/03	39,408.64	33,024.00
9	3002IL	1012	2004/03	129,024.48	110,760.00
10	3002IL	1022	2004/03	87,843.84	66,304.00

Note: You can also test the SQL by right-clicking the *SQL Object* and clicking *Expand*. Click the *Test SQL* tab to see a sample of the data.

Procedure: How to Test the SQL From the SQL Statement

1. Right-click the *SQL* object and click *SQL Statement*.

   The Select Statement dialog box opens, as shown in the following image.



   The Select Statement dialog box has the following buttons:

   - **Find.** Searches for a text string in the SELECT statement.
   - **Test SQL Statement.** Tests the SELECT statement.

   For more information on the shortcut menu options in this dialog box, see *Text Editor Shortcut Menu* on page 131.

2. Click the *Test SQL Statement* button in the upper-right corner.

   The Test SQL Statement dialog box shows a sample of the data.

Note: You can also test the SQL by right-clicking the SQL object, selecting *Column Selection*, and clicking the *Test SQL Statement* button in the Column Selection dialog box.

There is also a *Test SQL Statement* button in the Filter Calculator and the Sort Columns dialog box.

Procedure: How to Edit the SQL Statement

You can edit the generated SQL manually. If you do, the interface may redraw the SQL object based on the revised SQL when it is reloaded. If the interface is unable to redraw the SQL object, the SQL object will be labeled Custom SQL.

1. In the data flow workspace, right-click the *SQL* object and click *SQL Statement*.

   The Select Statement dialog box opens.

2. Make your changes to the SQL statement.
3. Click *OK* to save your changes.

Note: The SQL command must conform to the SQL supported by the server. For more information on using a dialect SQL specific to a relational database system, see *Creating a DBMS SQL Flow* on page 365.

Specifying Filters in a Data Flow

You can create a filter to control the data included in your data flow. A filter creates WHERE criteria in the SQL of your flow. You can filter your data with a constant value or a variable.

Note: When a filter is saved, it is parsed and any superfluous parentheses are removed.

Procedure: How to Add a Filter to a Data Flow

From the data flow workspace:

1. Right-click the SQL object and click *WHERE Filter*.

   The WHERE Filter Calculator opens.

2. Enter an expression in the Conditions window using the Columns/Variables and Functions tabs and the calculator buttons. For information about using functions, see *Using an SQL Function* on page 230. For information about using variables, see *Using Variables in a Flow* on page 314.
3. If you create multiple expressions that must all be satisfied, separate them with AND.
4. If you create multiple expressions where any of the expressions must all be satisfied, separate them with OR.
5. Click *OK*.

Procedure: How to Create a Data Flow Filter That Uses a Date/Time Constant Value

From the data flow workspace:

1. Right-click the SQL object and click *WHERE Filter*.

   The WHERE Filter Calculator opens.

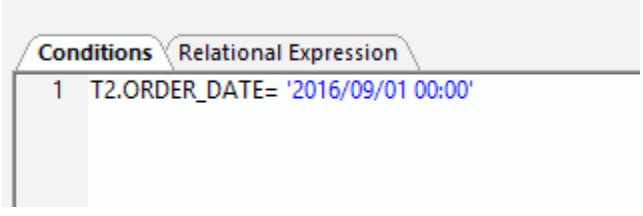
Note: If the Relational Expression tab is not selected, click on it to select it.

2. Double-click a column from the Columns/Variables tab. You can also type the column name in the Conditions field.

   The column is added to the Conditions field.

3. Use the calculator buttons to select a relationship.

4. Double-click the constant you want from the Date and Time Constants folder of the Columns/Variables tab, and replace the sample constant with the desired date/time. Character and date values must be enclosed in single quotation marks (‘), as shown in the following image.



5. Click *OK*.

Procedure: How to Create a Data Flow Filter That Uses a Constant Value

From the data flow workspace:

1. Right-click the SQL object and click *WHERE Filter*.

   The WHERE Filter Calculator opens.

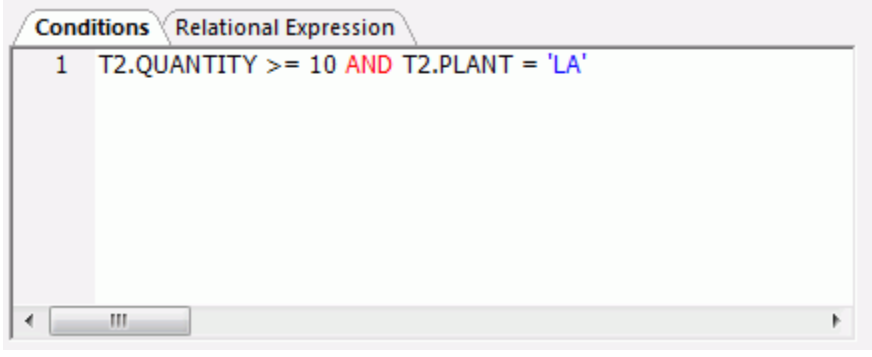
   Note: If the Relational Expression tab is not selected, click on it to select it.

2. Double-click a column from the Columns/Variables tab. You can also type the column name in the Conditions field.

   The column is added to the Conditions field.

3. Use the calculator buttons to select a relationship.

4. Type in the constant value. Character and date values must be enclosed in single quotation marks (‘).



5. Click *OK*.

Procedure: How to Create a Data Flow Filter That Uses a Variable

From the data flow workspace:

1. Right-click the SQL object and click *WHERE Filter*.

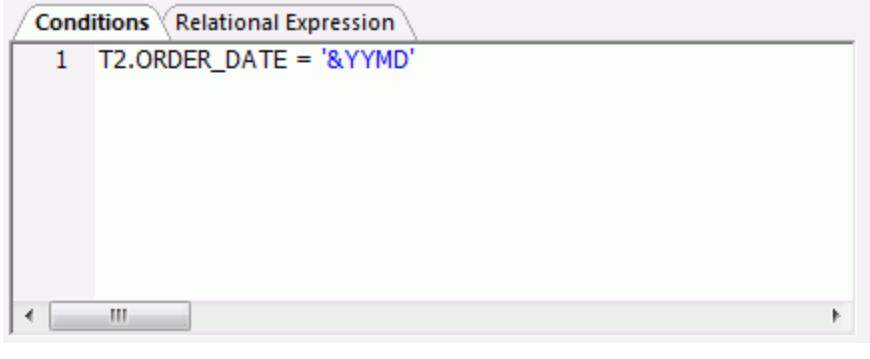
   The WHERE Filter Calculator opens.

2. Double-click a column from the Columns/Variables tab. You can also type the column name in the Conditions field.

   The column is added to the Conditions field.

3. Use the calculator buttons to select a relationship.

4. Double-click the variable you want from the DM or System Variables folder on the Columns/Variables tab. You can also type in character-valued or date-valued variables, which must be enclosed in single quotation marks (‘).



5. Click *OK*.

For more information about variables, see *Using Variables in a Flow* on page 314.

Procedure: How to Create a Filter That Uses a Sub-Select

A filter that uses a sub-select can be typed into the filter calculator or created using the Function Assist.

1. Drag *ibisamp/dmord* into a data flow.
2. Right-click the SQL object and click *WHERE Filter*.

   The WHERE Filter Calculator opens.

   Note: If the Conditions tab is not selected, click on it to select it.
3. Click the Functions tab, and expand *SQL Operators*.
4. Select the *IN ... SELECT* function.
5. A prototype condition is filled in:

   ```
   (test_exp1, test_exp2) IN (SELECT exp1, exp2 FROM mfd WHERE condition)
   ```
6. Replace the text with the desired condition. For example, select rows where the value of PLANT is found in DMSALE.

   ```
   T1.PLANT IN (SELECT T2.PLANT FROM  DMSALE T2 )
   ```

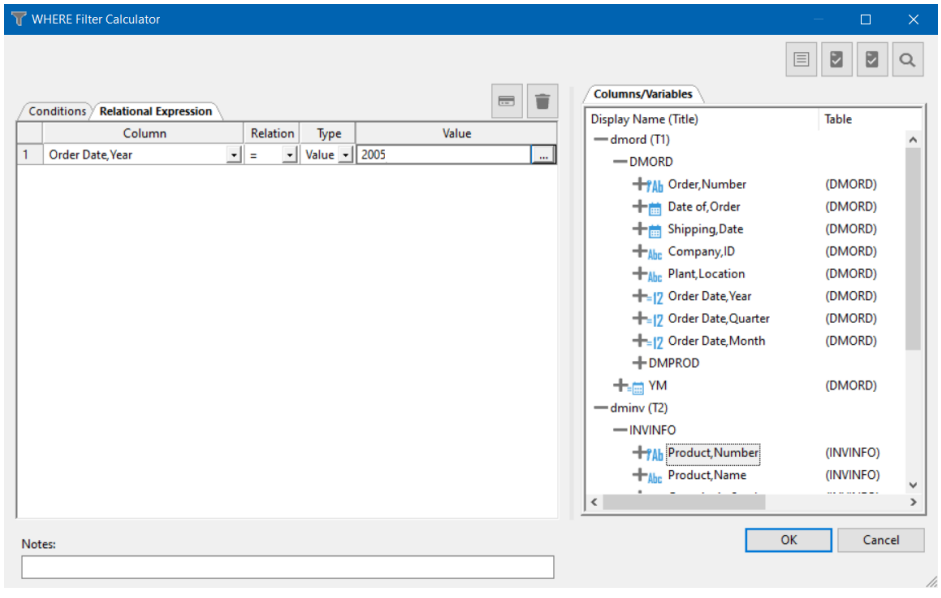
   You can also select rows where the combination of values of PLANT and ORDER_NUM are found in DMSALE.

   ```
   (T1.ORDER_NUM  ,T1.PLANT ) IN (SELECT T2.ORDER_NUM , T2.PLANT FROM
   DMSALE T2 )
   ```

Reference: WHERE Filter Calculator

To access the WHERE Filter Calculator from the data flow workspace, right-click the SQL object and click *WHERE Filter*.

The calculator opens, as shown in the following image.



The WHERE Filter Calculator contains the following fields and options. For a full description of the operators available, see *Join Calculator* on page 211.

The following options are available if you click the *Conditions* tab:

Conditions window

Displays the expression.

Function Assist button

Allows you to specify parameters for the function through a dialog box when creating or editing a transformation.

Calculator buttons

Insert numbers and operators.

( )

Adds parentheses.

''

Inserts two single quotation marks. Enter alphanumeric test values between these.

a->A

Converts selected text to uppercase.

A->a

Converts selected text to lowercase.

Date

Opens the Date Editor window, which lets you use the current date or specify a date from the calendar.

Datetime

Opens the Date Editor window, which lets you use the current date and time or specify a date and time from the calendar.

SQL buttons

The following SQL buttons are available:

- BETWEEN
- IN
- IS NULL
- LIKE
- AND
- OR
- ANY
- NOT
- ALL

Functions tab

Displays a list of SQL functions that are available for your transformations.

The following options are available if you click the *Relational Expression* tab:

Relational Expression window

Displays the Relational Expressions builder.



Test SQL Statement

Tests the SQL statement and displays the result.

Note: The test button will be grayed out if you have not selected any columns.

Columns/Variables tab

Displays available columns and variables in a hierarchical tree view or in a list view.

Columns

When displayed in a tree view, expanding the column name shows a list of values.

For a synonym for a data source in a table or file, a query is done to the data source to get a list of unique values. For large data sources, this could take a while.

A list of acceptable values is retrieved from the synonym if the synonym is for any of the following: database stored procedure, web service, application adapter, any other procedure based adapter.

Specifying Sorts

You can specify a sort to determine the order in which new columns appear in the results. A sort orders your results according to the results of a column you choose. When sorting by multiple columns, the first sort field sets the major sort break, the second sets the second sort break, and so on.

A sort is useful only in some cases. Use the following guidelines to consider whether or not to include a sort in your data flow.

- **FOCUS/FDS or XFOCUS data target.** A sort is useful for speeding up the process of loading the data target. Since loading without a sort will cause the process to take more time, you will receive a warning message.
- **Fixed format file.** A sort is useful if you are loading your data into an application that respects sorting. If it does not, a sort is not useful with a fixed format file.
- **Relational data target.** A sort is generally not useful because it is not respected when loading a relational data target.

Procedure: How to Specify a Sort

1. From the data flow workspace, right-click the *SQL* object and click *Sort*.

   The Sort Columns dialog box opens.

2. Double-click the column you want to sort on.

   or

   Select the column you want to sort on and click the right arrow.

   The column is added to the Sort Columns box.

3. Optionally, change the order in which the results will be sorted by clicking the drop-down menu under the Order column and selecting an order.

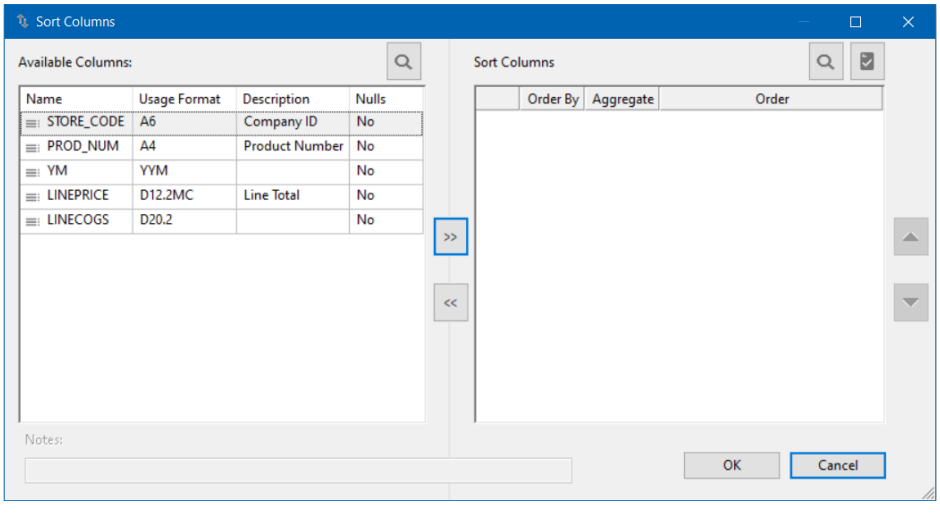
   - *ASC* (ASCENDING) sorts the results from lowest to highest. This is the default.
   - *DESC* (DESCENDING) sorts the results from highest to lowest.

4. Optionally, if you select more than one column, you can change the dominant sort by using the up and down arrows to move the columns up and down in the list.
5. Click *OK*.

Note: Clicking a column heading will sort the column grid on that heading.

Reference: Sort Columns Dialog Box

To access the Sort Columns dialog box from the data flow workspace, right-click the *SQL* object and click *Sort*.



The Sort Columns dialog box contains the following fields and options:

Available Columns

Is a list of columns available in the data source.

Source Column

Is the name of the column.

Format

Is the format of the column.

Description

Is a description of the column.

Note: Clicking a column heading will sort the column grid on that heading.

Nulls

Indicates if the column can contain NULL values.

Sort Columns

Is a list of sort columns.

Order By

Is the column(s) that sorts the results.

Order

Specifies whether results are sorted in ascending or descending order.

Aggregate

Is the column aggregation, if one has been applied.

Right and Left arrows

Add or remove sort columns.

 **Find**

Allows you to search for a column.

 **Test SQL Statement**

Tests the SQL statement and displays the result.

Note: The test button will be grayed out if you have not selected any columns.

Up and Down arrows

Move the selected sort column to change the order of the sort breaks.

Selecting a Data Target

You need to specify the data target(s) to which your data will be copied. You can choose to use an existing table as your target, or create a new target.

❑ For details on creating a new target, see *How to Create a New Data Target* on page 255.

Note: For new target tables, column descriptions are included in the synonym. The descriptions may come from the source table or from a source transform.

- For details on using an existing data target, see *How to Select an Existing Data Target* on page 261.

A data target can be a relational data target, non-relational data target, or a fixed format file. In some cases, relational targets can provide the ability to write to remote tables. For more information, see *Reviewing Your Data Targets* on page 94.

Note: This is only supported for existing targets. The name of the remote server must match the real server name of the subserver. This option should only be used for small data volumes. For large volumes, you should install the Server on the platform where the data will be loaded.

You can load your data into as many as fifteen tables in a single data flow.

After selecting a data target, you can do the following:

- Specify target options. For details, see *Specifying Data Target Options* on page 261.
- Specify mappings for existing tables. For details, see *Mapping Transformations* on page 305.

Reference: Data Target Types

The following data target types can be used:

- *Relational* data targets may include the following formats:
  - Access
  - Db2/UDB
  - EDA (for existing targets only)
  - Greenplum
  - Hive
  - Hyperstage
  - Informix
  - MySQL
  - Nucleus
  - ODBC
  - Oracle
  - Postgres
  - SQL Server
  - Sybase
  - Teradata

- *Non-relational* data targets may include the following formats:
  - FOCUS/FDS
  - XFOCUS
- *Other* data targets may include the following:
  - Delimited file (delimiter can be specified as a character string or hex digit)
  - XML or JSON file
  - Fixed format file
  - Formatted file
  - Hive Impala Phoenix/HBase MongoDB

Procedure: How to Create a New Data Target

1. Right-click in the space to the right of the SQL object, point to *Add Target*, and then click *New*.

   A target object appears in the workspace.

2. Right-click the new target and click *Properties*.

   The attributes of the target open in a new pane.

3. Select an Adapter from the drop-down menu to specify the target type.

   Note: Only data targets types that are configured on the server and supported by Data Migrator will appear.

4. Select a connection from the drop-down menu.
5. Select an Application Directory for the synonym that will be created for the target.
6. Optionally, enter a synonym name, which is the name of the data target.

   If you have chosen a FOCUS/FDS, XFOCUS, fixed format file, delimited file, or formatted file, when you press the Enter key, the data file name will automatically change to match the synonym name. This can be changed.

   Note: If you do not provide a synonym name, it will default to target01 for the first new target, target02 for the second, and so on.

   New targets will automatically contain the same columns that were selected in the SQL object. For information about managing columns, see *Selecting Columns* on page 220.

Note: If the columns selected from the source have descriptions, these descriptions will be included with the corresponding columns in a new target.

For relational database targets, you can specify connection (if applicable), application directory, synonym, and table name. For details, see *How to Set Options for Relational Targets* on page 293.

For FOCUS/FDS or XFOCUS files, you can specify synonym application directory, synonym, data application directory, and dataset (name of file). For details, see *How to Set Options for FOCUS/FDS or XFOCUS Targets* on page 297. Synonym names for FOCUS/FDS files cannot be longer than eight characters.

For fixed format files, you can specify synonym application directory, synonym, data application directory, and dataset (name of fixed format file). For formatted files, you can specify format, synonym application directory, synonym, data application directory, and dataset (name of formatted file). For details, see *How to Set Options for Fixed Format, Delimited, Formatted, and XML File Targets* on page 303.

Note: To create a file in HFS (hierarchical file system) on IBM z/OS, enter the name. To create a dataset enter the name as

```
//'qualif.tablename.type'
```

where:

`qualif`

Is the fully qualified location.

`tablename`

Should match the synonym name.

`type`

Is DATA or FOCUS, by convention.

7. Set the load options. For more information, see *Specifying Data Target Options* on page 261.
8. Click *OK*.
9. Right-click the target and click *Target Transformations*.
10. Select the key column(s) in the Target Columns grid.
11. Click *OK*.

Procedure: How to Create a New Data Target With a Variable Name

When creating a new target table, the name of the synonym that describes that table is normally entered as text and saved with the flow. This procedure should be used when you desire to specify the name of the target table dynamically when the flow is run.

1. Right-click in the space to the right of the SQL object, point to *Add Target*, and then click *New*.

   A target object appears in the workspace.

2. Right-click the new target and click *Properties*.

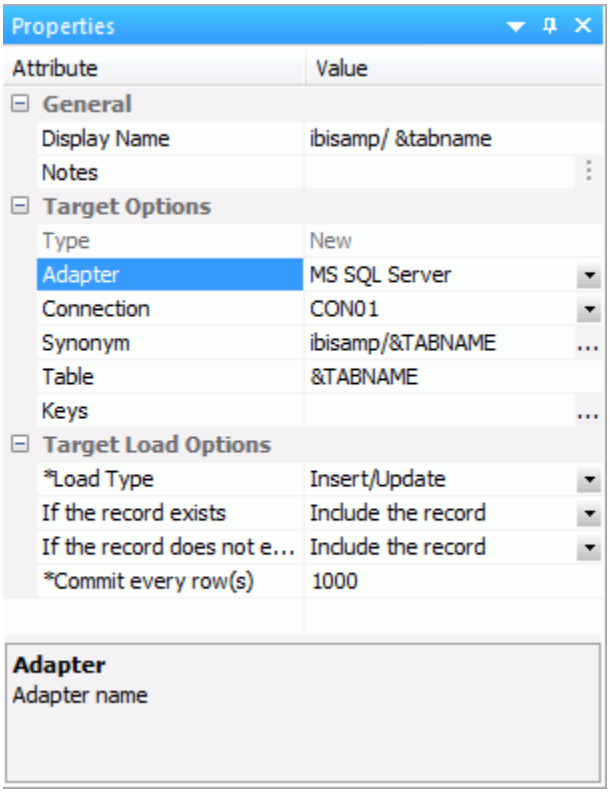
   The attributes of the target open in a new pane.

3. Select an Adapter from the drop-down menu to specify the target type.

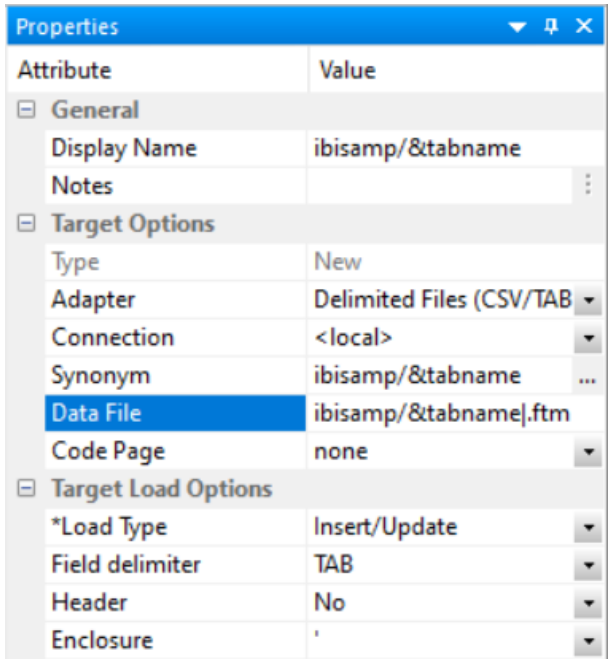
   Note: Only data target types that are configured on the server and supported by Data Migrator will appear.

4. For the synonym name, enter a name starting with an ampersand (&) to indicate that it is a variable.

The Properties pane for a MS SQL Server table target is shown in the following image.



The Properties pane for a Delimited File target is shown in the following image.



5. By default, the variable name you entered is automatically used as the name for the Table (relational database target) or Data File (fixed format file, delimited file, xml target). In the data file name, you must add an escape character (|) to concatenate the file name to the extension. If needed, another name can be used.
6. To test the data flow, click the Run button and click *Run*.

   A dialog box opens requesting a value for the variable.
7. Click *Run* to run the flow.

Procedure: How to Submit or Schedule a Data Flow With a Variable Name

To submit a data flow with a variable name, or to schedule such a flow, the value for the variable must be assigned. This can be done with a process flow that calls the data flow.

1. Right-click an application directory in the navigation pane, point to *New*, and then click *Flow*.

   The data flow workspace opens.
2. Click the *Process Flow* tab to open the process flow workspace.
3. Drag the data flow with the variable target name into the process flow workspace.

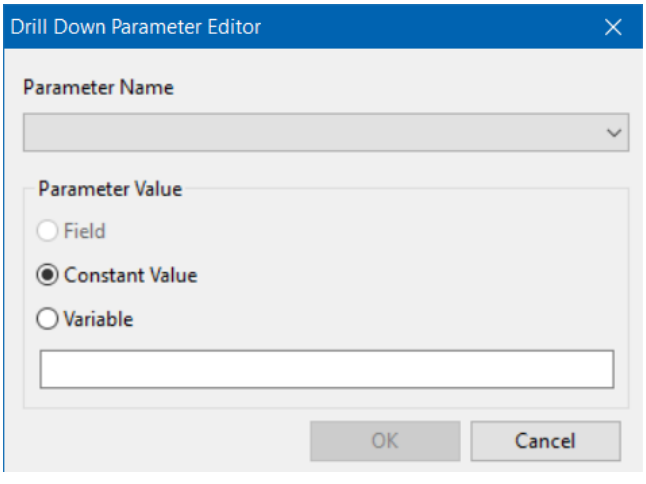
Right-click the *Start* icon, drag a line connecting to the data flow icon, and release. Then double-click the data flow icon to open its properties.

4. Click the ellipsis in the Parameters box.

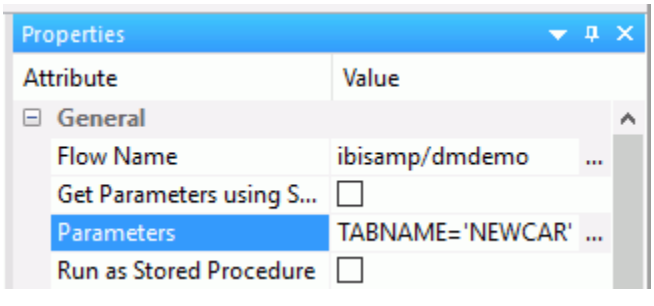
   The Parameter Editor opens.

5. Click the *Add Parameter* button  .

   The Drill Down Parameter Editor dialog box opens, as shown in the following image.



6. Select the name of the variable from the Parameter Name drop-down menu and enter a name for the table. Click *OK* twice to close the Parameter Editor dialog box. The parameter now appears in the properties window, as shown in the following image.



7. Click the *Save* button to save the data flow. The process flow can now be submitted, or scheduled, with the specified table name.

Procedure: How to Select an Existing Data Target

In the DMC, drag a synonym object from the navigation pane to the right of the SQL object in the workspace.

or

Right-click in the workspace to the right of the SQL object, point to *Add Target*, and then click *Existing*. Open an Application Directory, select an object, and click *Select*.

The data target is automatically connected to the SQL object.

Procedure: How to Change a Data Target in a Data Flow

With the data flow open in the workspace:

1. Right-click the target object that you want to change and click *Change Target*.

   The Select Source dialog box opens.

2. Select the replacement synonym and click the *Select* button.

Specifying Data Target Options

After the data targets have been added to the flow, there are two types of options that govern the behavior of the data flow when copying data into the targets.

- **Data flow properties**, which are set for the entire flow and, therefore, effect every data target. For convenience these options are displayed with each target. They are marked with an asterisk in the Target Properties dialog box to indicate that they are set at the flow level.
- **Target-specific properties**, which can be set on individual target objects.

Data flow properties include:

- **Load type**, which specifies the method Data Migrator uses to load the data. The options for load types depend on the target database type and may include: Insert/Update, Insert records from memory, Change Data Capture processing, Slowly Changing Dimension, Bulk load utility via disk file, and Extended Bulk Load Utility. If you are moving data to more than one data target, they will all use the same Load type. It will be displayed on all target objects and, if changed in one target, the change will be reflected in all the target objects. However, if you choose Insert/Update for the Load type, targets can have different settings for *If the record exists/does not exist*.

- **Record logging**, which allows you to create log files based on specific criteria. Each logging option can write information to a separate fixed format file. For example, if you choose Duplicate Rejection as a Record Logging option, a fixed format file called *procedurename*_dupl.ftm is created where duplicate records that are rejected will be recorded. You can use the log file as input for other processes. For details, see *How to Set the Data Flow Record Logging Options* on page 288.

- **Commit**, which specifies the number of rows to process before issuing a commit to a relational database target, or CHECK to a FOCUS or XFOCUS target. This option does not apply to file targets.

  A commit value of 0 indicates that a commit is only issued when the data flow completes.

  Note: A commit value of NONE can be used to indicate that Data Migrator should not issue any commit. This may be useful in cases where two or more flows should complete successfully before any data is committed to the target tables. A commit (or rollback) should be issued in a stored procedure after the data flows are complete. This option should be used with caution because if no commit or rollback is issued, the target table can become locked.

- **Blocksize**, which is the number of rows for loading in a block. This option only appears when insert records from memory is selected.

Target-specific properties include:

- **General Properties**, such as:

  - **Display Name.** The name that appears on the data flow under the target icon. By default, the synonym name and location (optionally) appear.

  - **Notes.** User entered notes. These notes only appear in the properties panel and now the data flow itself. To make the notes visible in the data flow, select the *Show Notes* option in the Data Flow Designer Options dialog box.

- **Prior to Load Options**, which can be set for existing targets to clear the target before the flow runs. The available options will depend on the type of data target.

- **If the record exists/does not exist** options, which specify behavior for Insert/Update.

Target Load Options

Load options specify the way Data Migrator loads data into the targets. If you are moving data to more than one target, they will all use the same load type. The load type is displayed on all target objects and, if changed in one target, the change will be reflected in all the target objects.

Load options vary depending on the type of target. They can include:

- **Insert/Update.** Allows you to set a behavior when loading records.

  If you select this option, you can set a behavior to apply when duplicate records are found. Select an option from the *If the record exists* drop-down menu. Insert/Update options can be set on a per target basis.

  If the record exists:

  - **Include the record.** Allows the relational database target to handle duplicates.
  - **Reject the record.** Issues a SELECT command against the target table to see if a record exists with the key values. If there are no key columns, Data Migrator screens all columns. If the record is found, the record is rejected.
  - **Update the existing record.** Updates records if the key value on the input record, or entire record if there are no keys, are found in the table. All non-key values are updated in the target.
  - **Delete the existing record.** Deletes the record if the key on the input record, or entire record if no keys, is found in the table.

  If the record does not exist:

  - **Include the record.** Includes the key in the data target.
  - **Reject the record.** Does not include the key in the data target. This option is only available if *Update* or *Delete the existing record* was selected.

    Note: When loading relational data targets, the Reject, Update, and Delete options can adversely affect performance, because they determine the existence of a key value on an incoming record *before* performing the specified action. This is done by issuing a SELECT command against the target table, and then waiting for the response. If one of these actions is required, try to limit the volume of records in the incremental change. These actions will perform best if there are unique keys on the table.

  If you select *Update the existing record* or *Delete the existing record* from the *If the record exists* drop-down menu, you can also set behavior when the record does not exist using the *If the record does not exist* drop-down menu:

  - **Commit every row(s).** Specifies the row interval to commit or write transactions to the database.

    When a new Flow uses the Optimized Load option:

    The *Commit every row(s)* parameter is not available in the Target Properties.

For Load Type *Slowly Changing Dimensions*, the commit is issued when the job is completed.

For Load Type *Bulk Load*, the commit is issued according to the adapter settings BULKLOAD and BULKCHECK.

- **Insert records from memory.** Speeds the loading of the data target by inserting a block of rows at once. You can set the row interval to commit or write transactions, and the number of records to load in a block.

  Note: This option is only available for Db2/UDB CLI, MySQL, MS SQL Server, Oracle, Informix, Sybase ASE, and Teradata 13. For other databases, the number of records in a block will default to 1 and cannot be changed. For details about loading relational targets, see *How to Set Options for Relational Targets* on page 293, and *Target Properties Pane for Relational Targets* on page 290.

  Insert records from memory:

  - Loads much faster, since a block of rows are inserted at once.
  - Requires input data to be clean. If any one row in the block causes a data source constraint violation, such as not null or unique index, the entire block is rejected for an MS SQL Server target. For other targets it depends on the database.
  - Does not provide row counts (the number of records inserted or rejected) in the detail log or statistics.
  - Does not provide record logging (used to write rejected records to a fixed format file for review).

- **Bulk load utility via a disk file.** Uses database bulk loaders instead of Data Migrator to insert data into a target. Data Migrator automates bulk loading for Greenplum, Hyperstage, Ingres, Informix, Microsoft SQL Server, IBM Db2, MySQL, Teradata, Nucleus, Oracle, Sybase Adaptive Server Enterprise, and Sybase Adaptive Server IQ. You can set format version, maximum number of errors, first and last row to copy, packet size, and row interval to commit or write transactions.

  The options available will depend on the RDBMS of your target type.

  Note that rows rejected by a DBMS vendor bulk load program are not counted as errors. Users of the bulk loader should look into the vendor specific documentation on how to handle the error handling of the bulk loader program.

Note:

- To use a database vendor bulk load program, such as *bcp* from MS SQL Server or Sybase, *sqlload* from Oracle, or *mysql* from MySQL, that program must be installed on the system where the Server is running and in the PATH.
- Does not provide row counts (the number of records inserted or rejected) in the detail log or statistics.

- **Extended Bulk Load Utility.** Uses a database loader program with additional capabilities. Currently available for Apache Hive, Cloudera Impala, Google BigQuery, EXASol, Greenplum, Hyperstage, Informix, Jethro, MongoDB, MS SQL Server ODBC, MySQL, Netezza, Oracle, PostgreSQL, Sybase ASE, Sybase IQ, Teradata, and 1010data.

  The options available will depend on the RDBMS of your target type. For more information on the options available for each adapter, see *Extended Bulk Load Utility Options* on page 266.

  Note: Does not provide row counts (the number of records inserted or rejected) in the detail log or statistics.

- **Change Data Capture Processing.** Enables you to load a data target with only the records that have changed. This feature is an optional, add-on component to Data Migrator available for selected relational databases.

- **Slowly Changing Dimensions.** Enables you to load a data target with column values that change over time. There are two column types. Type I simply overwrites the value. Type II allows you to track activity before and after the change. Type II changes are handled either using an activation flag or end date/begin date. When you update a Type II column, you do not actually change its value. Instead, you update the activation flag to inactive, or the end date to the current date. You then add a new record with the new value and the activation flag on by default or the begin date set.

- **Change Data Capture/Slowly Changing Dimensions.** A combination of the previous two load types. The Insert/Update/Delete indications in the CDC logs are not applied directly to the target. Instead, the SCD Types of the target are used.

- **Allow Direct Load to Parquet File.** Creates a file in the Hadoop Distributed File System (HDFS) in Parquet format, without first creating an intermediate text file. This option is available when using the Adapters for Hive and Impala.

Extended Bulk Load Utility Options

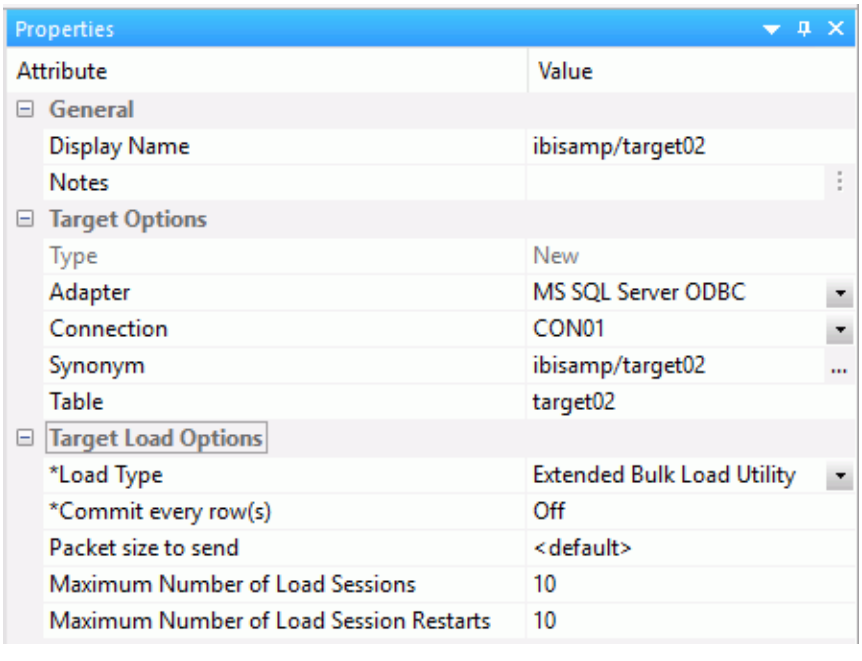
This option uses a database loader program with additional capabilities and is currently available for specific adapters. When extended bulk load is used for large data volumes, faster throughput is achieved by creating multiple load sessions (or threads). It reads from the source in parallel, a block at a time, where the block size is set by the *Commit every rows* parameter. It also creates multiple sessions for loading if the database load utility supports that functionality, up to the number specified in Maximum number of load sessions parameter.

Note:

- The options available will depend on the RDBMS of your target type.
- When Data Migrator creates a new synonym, it generates the columns in the metadata (Master File), in the same order as in the underlying table in the database. You should not manually change the order of the columns in the Master File.

  To use a synonym on a flow as a target with load type *Extended Bulk Load Utility Options*, ensure that the columns in the synonym are in the same order as they are in the underlying table. When the order of the columns in the Master file differs from the order of the columns in the table in the database, you should recreate the synonym or refresh it.

An example of the Extended Bulk Load Utility options for MS SQL Server can be seen in the following image:



The following option is available for every adapter that has an Extended Bulk Load Utility load type.

Commit every row(s)

Number of rows in a block. Use at least the default value of 1,000,000.

The following parameters depend on the database:

Maximum Number of Load Sessions

The maximum number of the concurrent load sessions logged on to the database. This parameter is applicable when the BULKLOAD parameter is set to *ON*. Its setting can be defined on the flow level and also on the Adapter level. The level setting of the adapter applies to the designs of the new Data Flows. The value set on a flow level takes precedence during the execution.

The default value is set to 10. However, the user should set the value carefully and consider current environment factors, such as the number of CPU cores, available memory, the parameters on the data access agents on a server, network capabilities, and the number of users running the requests.

Maximum Number of Load Session Restarts

The maximum number of restart attempts after recoverable load session error.

The above three options are the only Extended Bulk Load options available for the following adapters:

- EXASol
- Hyperstage
- MySQL
- MariaDB
- Netezza
- Oracle

Note:

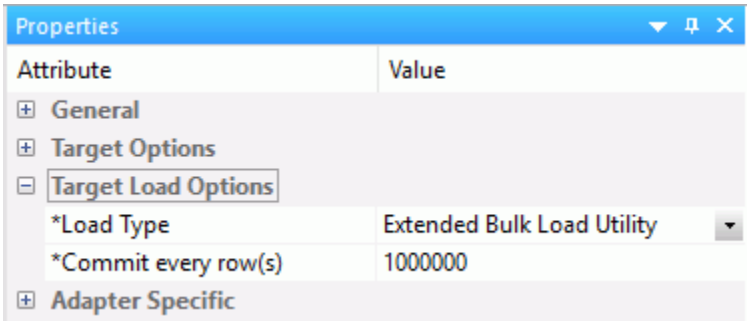
- The Extended Bulk Load Utility option must not be used on a target synonym that is for an Oracle table containing a Sequence. This is due to the rejection of triggers by the Oracle API used in the Extended Bulk Load Utility.

- The "Bulk thread postponed: too many connections" information message means that the maximum amount of loading threads was already started, so the new available intermediate load file will be queued and started when one of the loading threads become available. This situation occurs when either the Maximum number of Sessions value is too low or Commit value is too low.

The following sections describe the additional options that are available for each adapter.

Reference: **Apache Hive, Cloudera Impala, and Salesforce.com Extended Bulk Load Options**

targets only have the default Extended Bulk Load Utility option, as shown in the following image.



Note: For the best performance when loading Apache Hive or Cloudera Impala, it is recommended that the Server be located on a data node or Edge Node (with the Hadoop client installed) of your Hadoop cluster.

However, it is possible to use Data Migrator to load data into Hadoop, even when the servers are not co-located. In this case, the Server will create a temporary file and use FTP to copy it to the Hadoop node. This requires that an FTP server is available on the Hadoop cluster, and in addition for use with Impala REXEC daemon.

Example: **Configuring an Auxiliary Connection for Extended Bulk Load**

Using Extended Bulk Load with several adapters requires that intermediate data files are transferred to the server where the database is running or a data node of a Hadoop cluster. These adapters include:

- Apache Hive
- Apache Drill
- Cloudera Impala

- EXASol
- Jethro
- MongoDB
- 1010Data

For these adapters, you must configure Bulk Load to stage using FTP. From the DMC or Web Console, perform the following steps:

1. Expand the *Adapters* folder.
2. Expand the *Server Data Sources* folder.
3. Right-click the adapter connection and click *Configure Bulk Load*.

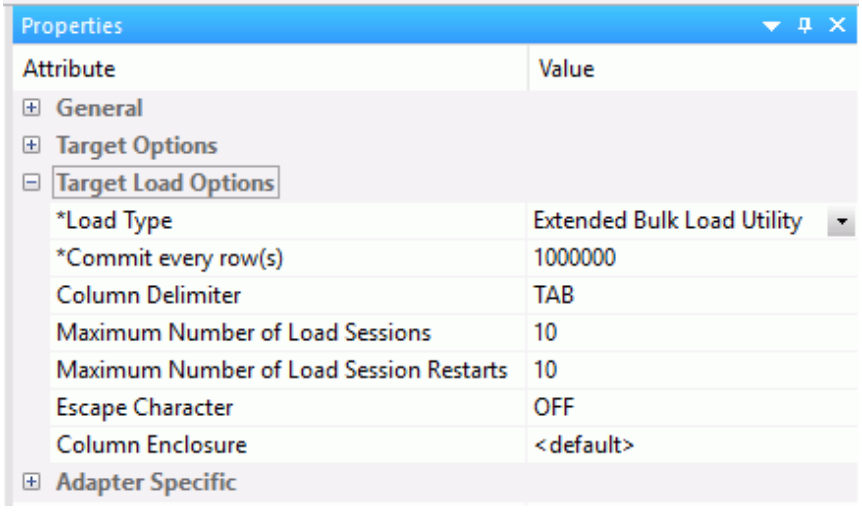
   The Configure Bulk Load dialog box opens.
4. From the Protocol drop-down menu, select *FTP* or *SFTP*.
5. In the Host Name box, type the name of the host for your server.

   Note: For Hive and Drill this must be a data node of your Hadoop cluster. For Impala, it must be a Data node or an Edge node. For Jethro, it must be the system where the Jethro server is running. For EXASol it should not be the system where it is running, rather it should be some other system that it can access using FTP.
6. In the Initial Directory box, type the name of a directory that you have write access to, such as your home directory.
7. In the User and Password boxes, type your user ID and password.
8. Click *Test*.

   The following message should appear: *User has read access to the FTP server*.
9. Click *OK* to close the dialog box.
10. If successful, click *Configure*.
11. Click *Close*.

Reference: Greenplum Extended Bulk Load Options

Greenplum targets have the following additional options, as shown in the following image.



Column Delimiter

Delimiter characters between fields in the intermediate fixed format file that is loaded into the database. The delimiter can be up to four characters and can be specified as the following:

- **TAB.** A tab character. This value is the default.
- **a - A.** A character string. For example, ~.
- **0x nn.** A hex code. For example, 0x44 (a comma), or 0x0D0A (a return and a linefeed). The hex code uses ASCII for Windows or UNIX systems and EBCDIC for IBM Mainframes.

Escape Character

Specifies the single character that is used for escape sequences, such as ,, or @;. Also used for escaping data characters that might otherwise be interpreted as row or column delimiters.

Column Enclosure

Characters used to enclose each alphanumeric value in the intermediate fixed format file that is loaded into the database. The enclosure consists of up to four printable characters. The most common enclosure is one double quotation mark. Numeric digits, symbols used in numbers, such as a period (.), plus sign (+), or minus sign (-), and 0x09 (Horizontal Tab), 0x0A (Line Feed), or 0x0D (Carriage return) cannot be used in the enclosure sequence. In order to specify a single quotation march as the enclosure character, you must enter four consecutive single quotation marks.

Note: When a non-default value is used, the intermediate file is created as the Greenplum term CSV, instead of what is called TEXT.

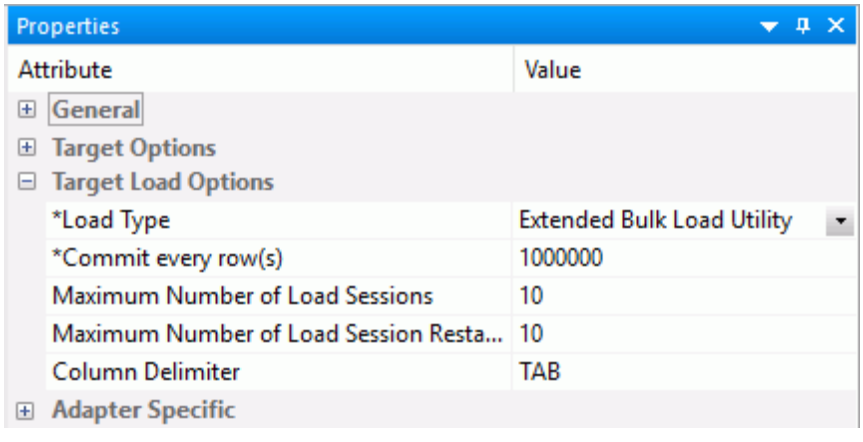
Global Settings

These options are only available when BULKLOAD=ON and apply to all new flows using Greenplum as the target.

Note: ALIAS_CASE controls the case of column names when new tables are created. Greenplum users may want to select *Enforce lower case*.

Reference: Informix Extended Bulk Load Options

Informix targets have the following additional options, as shown in the following image.



Column Delimiter

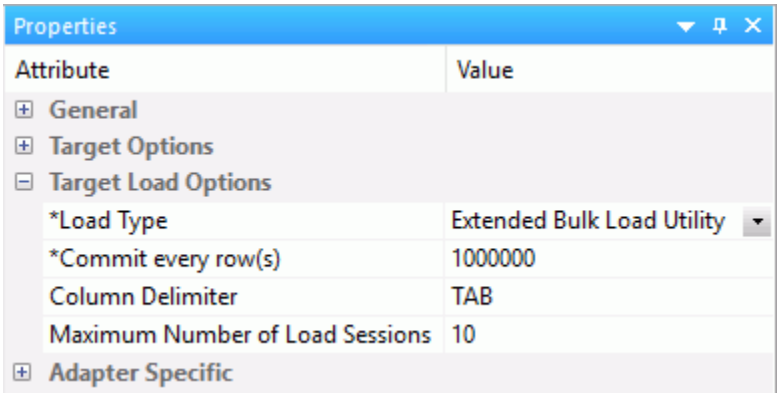
Delimiter characters between fields in the intermediate fixed format file that is loaded into the database. The delimiter can be up to four characters and can be specified as the following:

- **TAB.** A tab character. This value is the default.

- **a - A.** A character string. For example, ~.
- **0x nn.** A hex code. For example, 0x44 (a comma), or 0x0D0A (a return and a linefeed). The hex code uses ASCII for Windows or UNIX systems and EBCDIC for IBM Mainframes.

Reference: Jethro Extended Bulk Load Options

Jethro targets have the following additional options, as shown in the following image.



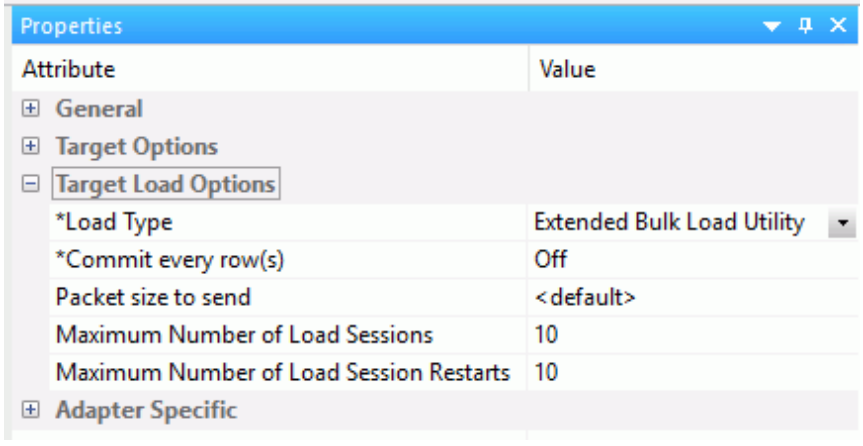
Column Delimiter

Delimiter characters between fields in the intermediate fixed format file that is loaded into the database. The delimiter can be up to four characters and can be specified as the following:

- **TAB.** A tab character. This value is the default.
- **a - A.** A character string. For example, ~.
- **0x nn.** A hex code. For example, 0x44 (a comma), or 0x0D0A (a return and a linefeed). The hex code uses ASCII for Windows or UNIX systems and EBCDIC for IBM Mainframes.

Reference: **MS SQL Server ODBC Extended Bulk Load Options**

MS SQL Server ODBC targets have the following additional options, as shown in the following image.

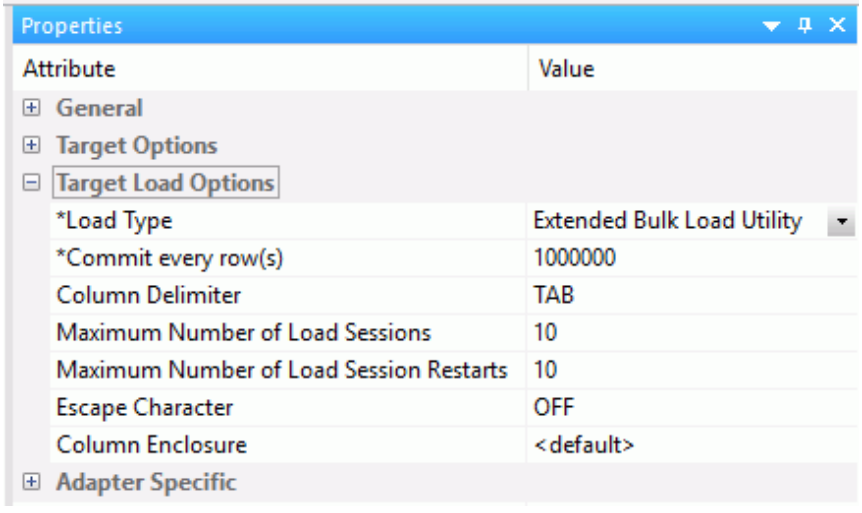


Packet size to send

The packet size in bytes.

Reference: PostgreSQL Extended Bulk Load Options

PostgreSQL targets have the following additional options, as shown in the following image.



Column Delimiter

Delimiter characters between fields in the intermediate fixed format file that is loaded into the database. The delimiter can be up to four characters and can be specified as the following:

- **TAB.** A tab character. This value is the default.
- **a - A.** A character string. For example, ~.
- **0x nn.** A hex code. For example, 0x44 (a comma), or 0x0D0A (a return and a linefeed). The hex code uses ASCII for Windows or UNIX systems and EBCDIC for IBM Mainframes.

Escape Character

Specifies the single character that is used for escape sequences, such as ,, or @;. Also used for escaping data characters that might otherwise be interpreted as row or column delimiters.

Column Enclosure

Characters used to enclose each alphanumeric value in the intermediate fixed format file that is loaded into the database. The enclosure consists of up to four printable characters. The most common enclosure is one double quotation mark. Numeric digits; symbols used in numbers, such as a period (.), plus sign (+), or minus sign (-); and 0x09 (Horizontal Tab), 0x0A (Line Feed), or 0x0D (Carriage return) cannot be used in the enclosure sequence. In order to specify a single quotation march as the enclosure character, you must enter four consecutive single quotation marks.

Note: When a non-default value is used, the intermediate file is created as the PostgreSQL term CSV, instead of what is called TEXT.

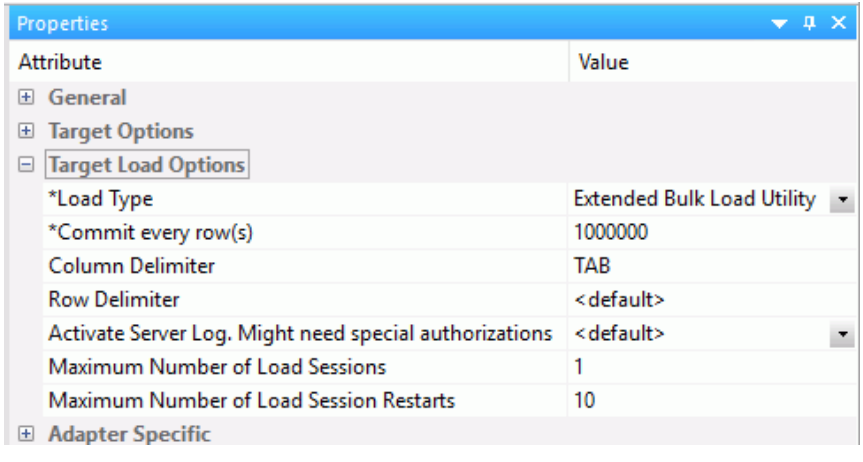
Global Settings

These options are only available when BULKLOAD=ON and apply to all new flows using PostgreSQL as the target.

Note: ALIAS_CASE controls the case of column names when new tables are created. PostgreSQL users may want to select *Enforce lower case*.

Reference: Sybase Extended Bulk Load Options

Sybase ASE and Sybase IQ targets have the following additional options, as shown in the following image.



Column Delimiter

Is the delimiter character(s) used between fields. It is used by the intermediate fixed format file used to load the database.

The delimiter can be up to four characters and can be specified as:

TAB. A tab character. This is the default value.

a - A. A character string. For example, ~.

0x nn. A hex code, for example, 0x44 (a comma) or 0x0D0A (a return and a linefeed).

Row Delimiter

Is the delimiter character used between records. It is used by the intermediate fixed format file to the load the database. The row delimiter can be specified in the same manner as the (field) delimiter, except that character comma (,) is not permitted.

Activate Server Log

This parameter is available for Sybase IQ only.

Activate log information about integrity constraint violations and the types of violations. With this option two types of LOG files could be written into a temporary space in the configuration directory: MESSAGE LOG file etlblk.msg and ROW LOG file etlblk.log.

For Sybase IQ:

- **Yes.** Writes both MESSAGE LOG and ROW LOG files: etlblk.msg and etlblk.log.
- **No.** The LOG files are not written. This is the default value.

Note:

- Extended Bulk Load supports all Sybase IQ native data types, except BINARY and VARBINARY.
- A user running a data flow using Extended Bulk Load to Sybase IQ must have the following permissions:

```
allow_read_client_file = on
allow_write_client_file = on
```

This can be confirmed by using the following Sybase stored procedure:

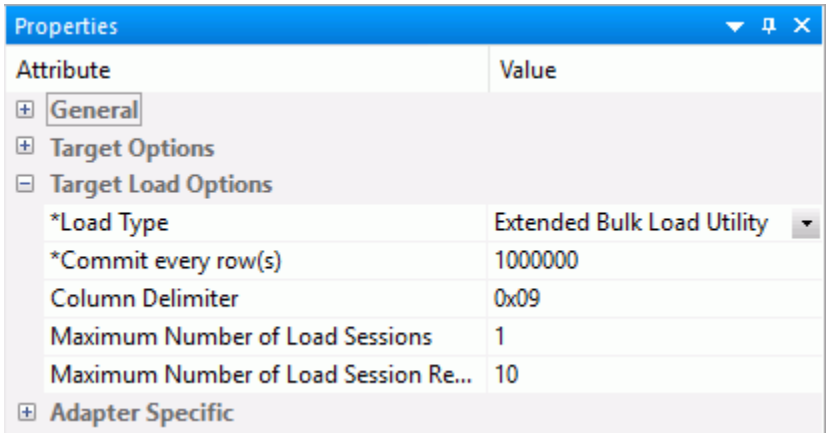
```
sp_iqcheckoptions
```

If these permissions are not set, the flow will fail and display the following error:

```
FOC1261 PHYSICAL INTERFACE NOT FOUND FOR SUFFIX SQLSYB. MODULE
NAME : dbodbc12_r
```

Reference: 1010 Extended Bulk Load Options

1010 targets have the following additional options, as shown in the following image.



Column Delimiter

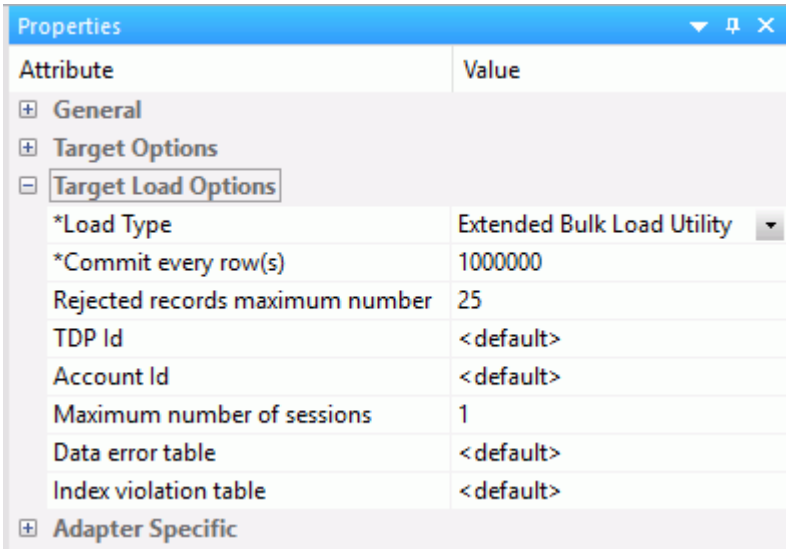
Delimiter characters between fields in the intermediate flfixed formatat file that is loaded into the database. The delimiter can be up to four characters and can be specified as the following:

- **TAB.** A tab character. This value is the default.
- **a - A.** A character string. For example, ~.
- **0x nn.** A hex code. For example, 0x44 (a comma), or 0x0D0A (a return and a linefeed). The hex code uses ASCII for Windows or UNIX systems and EBCDIC for IBM Mainframes.

Reference: Teradata Extended Bulk Load Options

The Teradata Extended Bulk Load Utility uses TPT (Teradata Parallel Transporter) to load data.

Teradata targets have the following additional options, as shown in the following image.



Rejected records maximum number

The maximum number of rejected records that can be stored in one of the error tables before the job is terminated. If unspecified, the default is unlimited.

TDP Id

The Teradata Director Program Id.

Account Id

The Account Id used to access the database.

Data error table

Name of table to contain records that were rejected because of data conversion errors, constraint violations, or AMP configuration changes. This must be a new table name. The default is the database default.

Index violation table

Name of table to contain records that violated the unique primary index constraint. This must be a new table name. The default is the database default.

Using Direct Bulk Load for a Flow

Direct Bulk Load is a feature of Data Migrator and Data Flow that can significantly shorten load times when a source file is in the expected format of the bulk load program for the database to be loaded. The faster load times result from loading the file directly, without creating an intermediate file.

Direct Bulk Load can be enabled as an adapter setting, meaning that all flows using that adapter use it. This can be problematic if some input files met the requirements of a database and some did not.

You can enable or disable Direct Bulk Load on each data flow. However, in order for Direct Bulk Load to be enabled for a flow, Extended Bulk Load must be used and Optimize Load must be enabled.

When Extended Bulk Load is used and Optimize Load is enabled, data is loaded directly to the target database without the additional extract and transformation steps provided the following conditions are met:

- The data source is a delimited file.
- The file attributes including Header, Enclosure, Delimiter and Record Delimiter are supported by the target database.
- Only insert operations are used (no updates).
- No filters are used.
- No aggregation is specified.
- No transformations are used.

When Direct Bulk Load is used, the following message appears in the Data Migrator log:

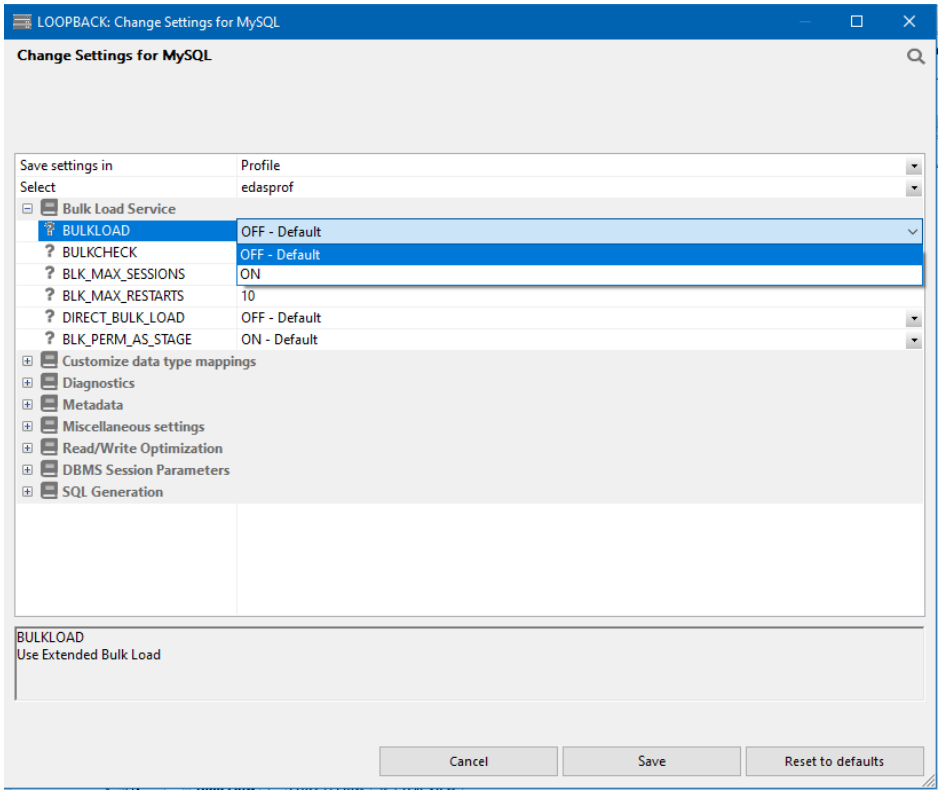
```
(ICM18637) Load Operation will use Direct Bulk Load Feature
```

The following are the supported databases and data stores:

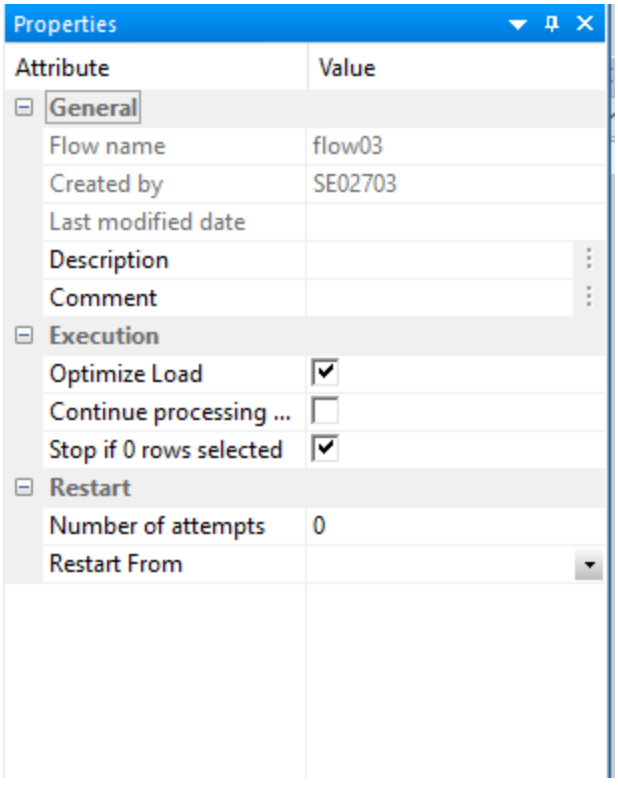
- Apache Hive
- Apache Impala
- Apache Spark
- Amazon RedShift
- ExaSOL

- Jethro
- MariaDB
- MS SQL Server
- MongoDB
- MySQL
- PostgreSQL
- Snowflake
- Salesforce.com
- Sybase ASE
- Vertica

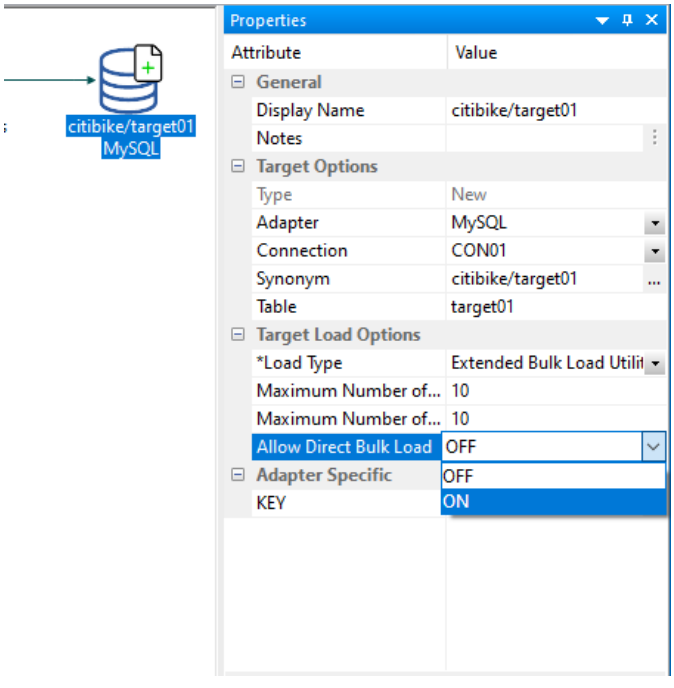
The following Change Settings for MySQL page shows Extended Bulk Load being enabled.



The following image shows Optimize Load enabled in the Flow Properties (by right-clicking the canvas to the right of the SQL object and selecting Flow Properties).



The following image shows the option to enable Direct Bulk Load in the Load Options for a flow.



Reference: Db2 Target Bulk Load Options

Db2 targets have the following options:



Loading method

INSERT. Insert the rows only. This is the default.

REPLACE. Drops the existing destination table, and creates a new table.

Note:

- The bulk load option for Db2 is available for z/OS and IBM i only when configured for CLI, not CAF.

- For bulk load on IBM i, the target table name for Db2 must not exceed 10 characters.
- For bulk load on IBM i, the target table name must be a qualified name. The current library name must be specified, for example, library1/table1.

Reference: Db2 Target Insert Records from Memory Options

The Bulk Insert API (FASTLOAD) could be in effect for Db2 targets with load option Insert records from memory. The command that changes the settings for the Db2 FASTLOAD is:

```
ENGINE DB2 SET FASTLOAD [ON|OFF]
```

If effective, the default value for FASTLOAD is ON.

A SET command for Db2 to control whether or not the FASTLOAD operations are recoverable is:

```
SQL DB2 SET FASTLOAD_OPTION RECOVERABLE/NONRECOVERABLE
```

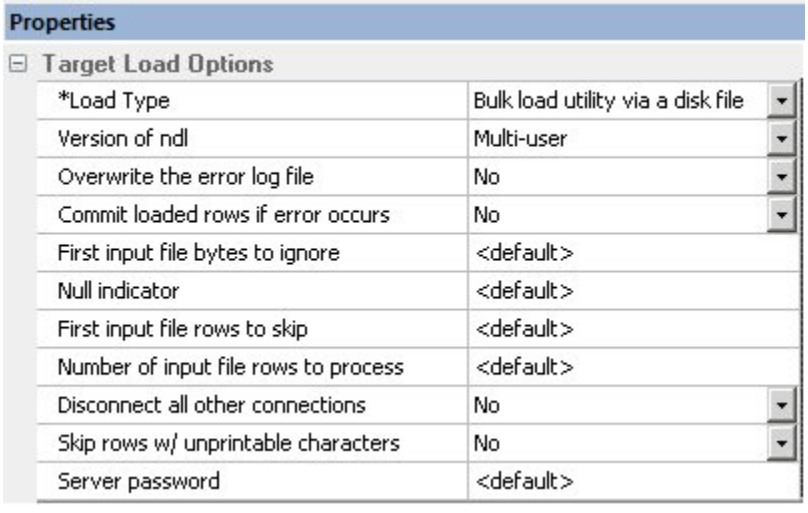
The default value is RECOVERABLE.

Reference: MySQL Bulk Load Options

There are no additional bulk load options for MySQL.

Reference: Nucleus Target Bulk Load Options

Nucleus targets have the following options:



Version of ndl

Single-user

Starts the server in exclusive mode. No other database sessions may be established.

Multi-user

The server must already be started in multi-user mode and listening on the default port 1285.

Overwrite the error log file

Yes

Creates a new file.

No

Appends to existing log.ndl file.

Commit loaded rows if error occurs

Yes

Issues a commit.

No

Issues a rollback.

First input file bytes to ignore

Number of bytes to skip before loading data.

Null indicator

Specifies the character to represent null.

First input file rows to skip

Number of rows to skip before loading data.

Number of input file rows to process

Number of rows to process before stopping.

Disconnect all other connections

Disconnects connections.

Skip rows w/ unprintable characters

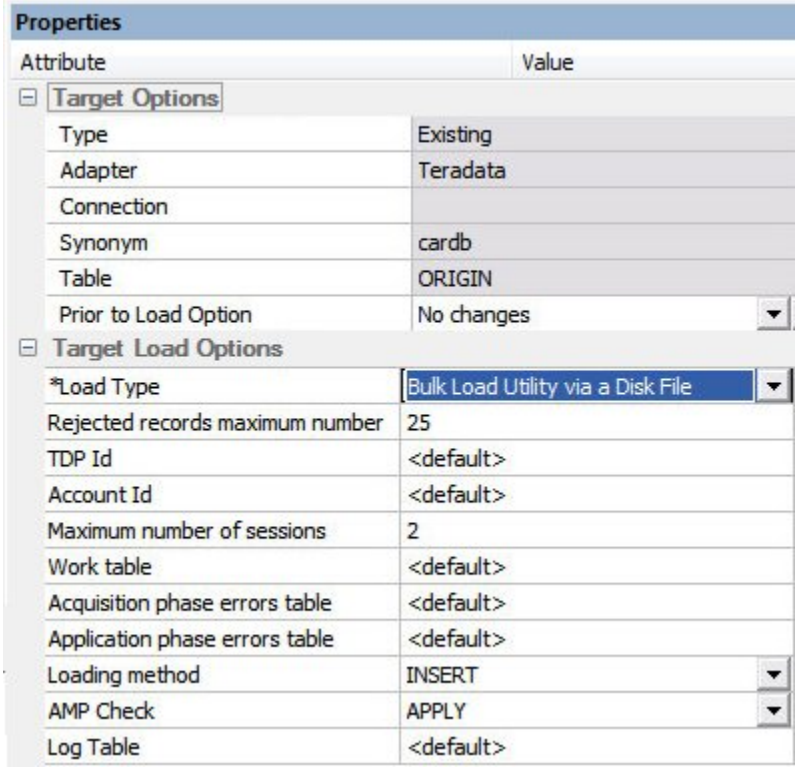
Skips rows with unprintable characters.

Server password

NDL server password. Required if Single User (ndls) mode and a server password actually exists.

Reference: Teradata Target Bulk Load Options

Teradata targets have the following options:



Note: Teradata has two bulk load programs that Data Migrator supports. FastLoad is used for new target tables, and MultiLoad is used for existing target tables. Not all options shown in the image above are available for new target tables.

UNIX only

Two environment variables, $FASTLOAD_EXE and $MLOAD_EXE, should be set for using bulk load.

$FASTLOAD_EXE. Specifies the location of the Teradata FastLoad utility.

$MLOAD_EXE. Specifies the location of the Teradata MultiLoad utility.

Two lines should be added to the profile, or server start up shell script (edastart), that are similar to the following:

```
export FASTLOAD_EXE = ~teradata/fastload
export MLOAD_EXE = ~teradata/mload
```

Rejected records maximum number

The maximum number of rejected records allowed before loading stops. The default is the target RDBMS default.

TDP Id

The Teradata Director Program Id.

Account Id

The account Id used to access the database.

Maximum number of sessions

The maximum number of MultiLoad or FastLoad sessions logged on to the Teradata database.

Start record number in source

Row number to begin copying data to the target.

End record number in source

Row number to end copying data to the target.

Work table

Name of the work table.

Acquisition phase errors table

This table provides information about all errors that occur during the acquisition phase of your Update operator job, as well as some errors that occur during the application phase if the Teradata RDBMS cannot build a valid primary index.

Application phase errors table

This table provides information about uniqueness violations, field overflow on columns other than primary index fields, and constraint errors.

Loading Method

INSERT. Inserts the rows only. This is the default.

UPSERT. Does inserts for missing update rows.

Note: This option is available for existing target tables with keys.

AMP Check

The MultiLoad response to a down Access Module Processor (AMP) condition.

ALL. Pauses the MultiLoad job when AMP is down.

NONE. Allows the MultiLoad job to start, restart, or continue as long as no more than one AMP is down in a cluster.

APPLY. Inhibits the MultiLoad job from entering or exiting the application phase when an AMP is down.

Note: This option is available for existing targets only.

Log Table

Restarts the log table for the MultiLoad checkpoint information. This option is available for existing target tables only.

Notes

Enter an annotation for this target.

Procedure: How to Set the Data Flow Record Logging Options

Setting the data flow record logging options allows you to write particular types of transactions to log files.

Although Record Logging options appear in every data target, they are set on a per flow basis. Changing the options in one target will reset them for all targets.

1. On the Flow tab, in the Tools group, click *Properties*, or right-click anywhere in the workspace and click *Flow Properties*.

   The Properties pane opens.

2. Expand the *Record Logging* attribute.
3. Select the check box next to the options that you want to log.

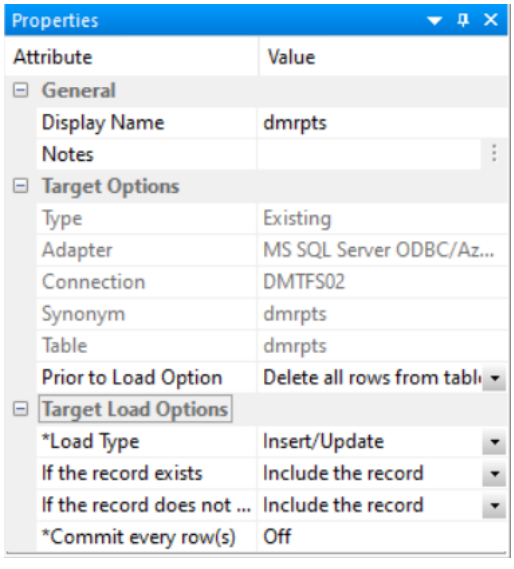
   Record logging options are optional and are not available when the data target is a formatted file.

4. Click *Save* in the Quick Access Toolbar.

   For more information on record logging options for a data flow, see *How to Set the Flow Record Logging Properties* on page 150.

Target Properties Pane

To access the Target Properties pane, right-click a target object and click *Properties*. The following image shows the Properties pane for an existing Relational target.



All targets have the following properties:

Display Name

By default, shows the application name, if enabled, and synonym name. Controls the name that is shown under the object in the flow.

Notes

Enter an annotation for this target.

Type

Is the target type.

Adapter

Is the target adapter.

Synonym

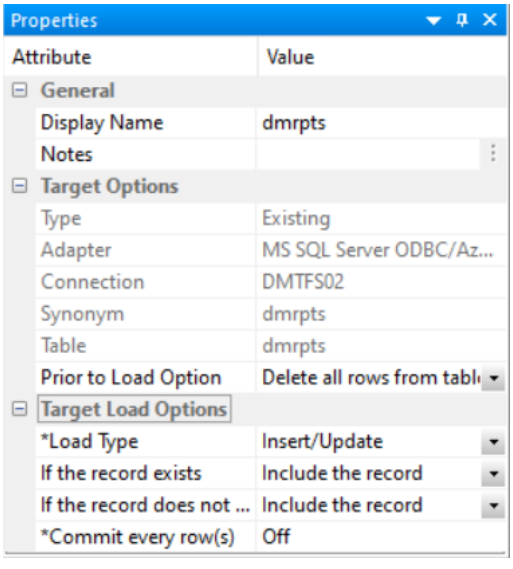
Is the synonym name.

Note: Synonym names cannot be longer than eight characters.

The following sections describe the Target Options available for each different target.

Reference: Target Properties Pane for Relational Targets

The Target Options section of the Properties pane for relational targets contains the following additional options:



Connection

Is the connection for the data target. For a relational data target, this is a database server. For ODBC, this is a data source.

Table

Is the name of data target table.

Keys

Is the names of key columns in order. This option is only available for new targets.

Prior to Load Options:

These options are only available for existing targets.

No changes

Does not delete the records already in a data target.

Delete all rows from table

Deletes all rows and creates a database log.

Truncate table

Deletes all rows from the table, but does not generate a database log. This is a faster option than using Delete all rows from table.

The Target Load Options include the following load options:

Load Type:

Note: Although Load Type appears in every data target, it is set on a per flow basis. Changing the type in one target will reset it for all targets.

Insert/Update

Specifies the behavior of Data Migrator while loading data.

Insert/Update options can be set on a per target basis.

If you select this option, you can set a behavior when duplicate records are found from the If the record exists drop-down menu.

- **Include the record.** Allows the relational database target to handle duplicates.

  Note: If you select *Include the record*, the record is passed directly to the relational database, which determines whether to accept it or not. If inserting a record would result in a duplicate key, the RDBMS will reject it due to a Unique Index constraint violation and return an error. Processing continues even if such errors occur, up to the number of errors specified under *Stop processing after __ DBMS errors* in the General properties of the flow.

- **Reject the record.** Issues a SELECT command against the target table to see if a record exists with the key values. If there are no key columns, Data Migrator screens all columns. If the record is found, the record is rejected.

- **Update the existing record.** Updates records if the key value on the input record, or entire record if there are no keys, are found in the table. All non-key values are updated in the target.

- **Delete the existing record.** Deletes the record if the key on the input record (or entire record if no keys) is found in the table.

  If you select *Update the existing record* or *Delete the existing record* from the If the record exists drop-down menu, you can also set behavior when the record does not exist using the If the record does not exist drop-down menu:

- **Include the record.** Includes the key in the data target.

- **Reject the record.** Does not include the key in the data target.

  Note: The Reject, Update, and Delete options can adversely affect performance because they determine the existence of a key value on an incoming record *before* performing the specified action. This is done by issuing a SELECT command against the target table, then waiting for the response. If one of these actions is required, try to limit the volume of records in the incremental change. These actions will perform best if there are unique keys on the table.

- **Commit every row(s).** Specifies the row interval to commit or write transactions to the database.

Insert records from memory

When you specify this load type, you need to specify:

- **Commit every row(s).** Specifies the row interval to commit or write transactions to the database.

- **Block size.** Specifies how many records you want to process at a time.

Note: The value for Commit every rows should be a multiple of the Block size, for example, 10000 and 2000.

Bulk load utility via a disk file

These options will depend on your target type.

Extended bulk load utility

Uses a database loader program with additional capabilities. For more information, see *Extended Bulk Load Utility Options* on page 266.

Slowly Changing Dimensions

This option is only available for existing relational targets.

Enables you to load a data target with column values that change over time. There are two column types. Type I simply overwrites the value. Type II allows you to track activity before and after the change. Type II changes are handled either using an activation flag or an end date/begin date. When you update a Type II column, you do not actually change its value. Instead, you update the activation flag to inactive, or the end date to the current date. You then add a new record with the new value, and the activation flag on by default or the begin date set.

- **Updates for Type I.** Enables you to specify how updates will affect Type I fields. The options are Change all rows or Change only active rows. Change all rows is the default value.
- **Commit every row(s).** Specifies the row interval to commit or write transactions to the database.

Change Data Capture Processing

Enables you to load a data target with only the records that have changed. This feature is an optional, add-on component to Data Migrator. When this option is selected, the *Prior to Load* option is eliminated.

Change Data Capture/Slowly Changing Dimensions

This load type is a combination of the previous two load types. The Insert/Update/Delete indications in the CDC logs are not applied directly to the target. Instead, the SCD Types of the target are used and deleted records in the source are marked as inactive in the target.

Procedure: How to Set Options for Relational Targets

1. In the data flow workspace, right-click the data target, and click *Properties*.

   The target Properties pane opens.

2. For existing targets, select whether to remove data prior to loading the data target in the Prior to Load Option drop-down menu. The options are:

   - **No Changes.** Does not delete the records already in a data target.
   - **Delete all rows from table.** Deletes all rows and creates a database log.
   - **Truncate table.** Deletes all rows from the table but does not generate a database log. This is a faster option than Delete all rows from table.

     Note: Truncate table is not supported by Db2 on IBM i.

3. Select a Load Type. The options are:

   - **Insert/Update.** Allows you to set a behavior when loading records.
   - **Insert records from memory.** Speeds the loading of the data target by inserting a block of rows at once. You can set the row interval to commit or write transactions and the number of records to load in a block. This option is only available for relational databases that support it, including Db2 on i V6R1 for CLI, Db2 on z, Informix, MS SQL Server, MySQL, Oracle, Sybase ASE, Teradata 13, and UDB.

     This option:

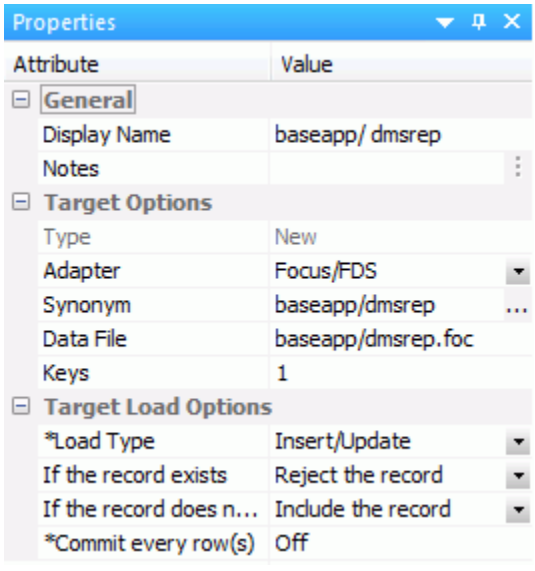
     - Requires clean input data. If any row in the block causes a data source constraint violation, such as not null or unique index, the entire block is rejected.
     - Does not provide row counts (the number of records inserted or rejected) in the detail log or statistics. NA (not available) will appear instead.
     - Does not provide record logging. For example, rejected records cannot be written to a fixed format file for review.
   - **Bulk load utility via a disk file** and **Extended Bulk Load Utility**. Use database bulk loaders instead of Data Migrator to insert data into a target. Data Migrator automates bulk loading for Hyperstage, Ingres, Informix, Microsoft SQL Server, IBM Db2, Teradata, Nucleus, Oracle, Sybase Adaptive Server Enterprise, and Sybase Adaptive Server IQ. You can set format version, maximum number of errors, first and last row to copy, packet size, and row interval to commit or write transactions.
   - **Slowly Changing Dimensions.** Enables you to load a data target with column values that change over time. There are two column types. Type I simply overwrites the value. Type II allows you to track activity before and after the change. Type II changes are handled using either an activation flag or an end date/begin date. When you update a Type II column, you do not actually change its value. Instead, you update the activation flag to inactive, or the end date to the current date by default. You then add a new record with the new value and the activation flag on by default or the begin date set.
   - **Change Data Capture Processing.** Enables you to load a data target with only the records that have changed. This feature is an optional, add-on component to Data Migrator.

     Note: The Change Data Capture Processing option is only available for existing relational targets.

- **Change Data Capture/Slowly Changing Dimension.** Is a combination of the previous two load types. The Insert/Update/Delete indications in the CDC logs are not applied directly to the target. Instead, the SCD Types of the target are used.

Reference: Target Properties Pane for FOCUS/FDS or XFOCUS Targets

The Target Options section of the Properties pane for FOCUS/FDS or XFOCUS targets contains the following additional options:



Data File

Is the name of the data file pointed to by the synonym.

On IBM z/OS, to create a file in HFS (hierarchical file system), enter the name. To create a dataset, enter the name as

```
//'qualif.tablename.FOCUS'
```

where:

qualif

Is a fully qualified location.

tablename

Should match the synonym name.

Keys

Is the number of key columns. This option is only available for new targets.

Prior to Load Options:

These options are only available for existing targets.

No changes

Does not drop (delete) the data target.

Drop Table

Drops and recreates the data target.

Load Type

Specifies the method Data Migrator uses to load data.

Insert/Update

Since Data Migrator uses Insert/Update to load FOCUS/FDS or XFOCUS targets, you can set a behavior when duplicate records are found from the If the record exists drop-down menu. These options can be set on a per target basis.

- **Include the record.** Includes the duplicate record in the data target.
- **Reject the record.** Issues a SELECT command against the target table to see if a record exists with the key values. If there are no key columns, Data Migrator screens all columns. If the record is found, the record is rejected.
- **Update the existing record.** Updates records if the key value on the input record, or entire record if there are no keys, are found in the table. All non-key values are updated in the target.
- **Delete the existing record.** Deletes the record if the key on the input record (or entire record if no keys) is found in the table.

  If you select *Update the existing record* or *Delete the existing record* from the If the record exists drop-down menu, you can also set behavior when the record does not exist using the If the record does not exist drop-down menu:

  - **Include the record.** Includes the key in the data target.
  - **Reject the record.** Does not include the key in the data target.
  - **Commit every row(s).** Specifies the row interval to commit or write transactions to the database.

Procedure: How to Set Options for FOCUS/FDS or XFOCUS Targets

1. In the data flow workspace, right-click the data target and click *Properties*.

   The Target Properties pane opens.

2. For existing targets, select whether to remove data prior to loading the data target in the Prior to Load Options section. The options are:

   - **No changes.** Does not delete the records already in a data target.
   - **Drop Table.** Drops and recreates the data target.

3. Select a behavior for loading records using the If the record exists and If the record does not exist drop-down menus. These options can be set on a per target basis. They are:

   - **Include the record.** Includes the duplicate record in the data target.
   - **Reject the record.** Issues a SELECT command against the target table to see if a record exists with the key values. If there are no key columns, Data Migrator screens all columns. If the record is found, the record is rejected.
   - **Update the existing record.** Updates records if the key value on the input record, or entire record if there are no keys, are found in the table. All non-key values are updated in the target.
   - **Delete the existing record.** Deletes the record if the key on the input record (or entire record if no keys) is found in the table.

4. If you select *Update the existing record* or *Delete the existing record* from the If the record exists drop-down menu, you can also set behavior when the record does not exist using the If the record does not exist drop-down menu:

   Include the record includes the key in the data target.

5. Click *OK*.

   Note:

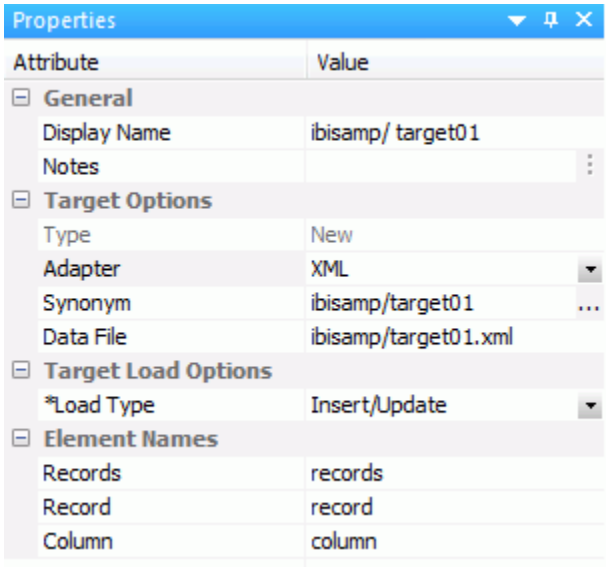
   - For FOCUS/FDS or XFOCUS targets, the only Load Type available is Insert/Update.
   - Synonym names for FOCUS/FDS files cannot be longer than eight characters.

Reference: Target Properties for New XML Targets

Data Migrator can create a new XML document with name/value pairs. For more information on any other structures, see *Creating a Data Flow Using a Target Based on a Predefined XML Schema* on page 590.

Note: In order to create an XML document, a Server Administrator must first configure an adapter for XML.

The Target Options section of the Properties pane for XML targets contains the following additional options:



Data File

Is the name of the XML document described by the synonym. Data Migrator will also create an XML Schema Definition with the same name as the data file and an extension of .xsd.

Load Type

Is set to Insert/Update for XML documents. Note that no updates are currently performed.

Records

Is the name of the top level element.

Record

Is the name of the row level element.

Column

Is the name of the column level element.

Example: **New XML Targets**

The following example uses the same table ibisamp/dminv as a source with all the actual columns, and then creates a new XML target called dminvx. The element names are entered as shown:

Records - Inventory

Record - Item

Column - Detail

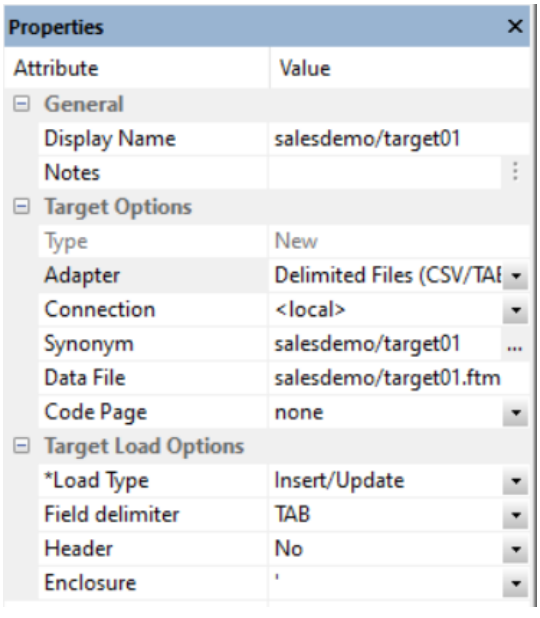
The first three rows of the resulting XML document would look like the following example when viewed in a browser:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <inventory xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <item>
      <detail name="PROD_NUM">1004</detail>
      <detail name="PRODNAME">2 Hd VCR LCD Menu</detail>
      <detail name="QTY_IN_STOCK">43068</detail>
      <detail name="PRICE">179.00</detail>
      <detail name="COST">129.00</detail>
    </item>
  - <item>
      <detail name="PROD_NUM">1006</detail>
      <detail name="PRODNAME">Combo Player - 4 Hd VCR + DVD</detail>
      <detail name="QTY_IN_STOCK">13527</detail>
      <detail name="PRICE">399.00</detail>
      <detail name="COST">289.00</detail>
    </item>
  - <item>
      <detail name="PROD_NUM">1008</detail>
      <detail name="PRODNAME">DVD Upgrade Unit for Cent. VCR</detail>
      <detail name="QTY_IN_STOCK">199</detail>
      <detail name="PRICE">199.00</detail>
      <detail name="COST">139.00</detail>
    </item>
```

Reference: **Target Properties Pane for Delimited File Targets**

The Target Options section of the Properties pane for delimited file targets contains the following additional options:

Note: Delimited File will only appear as a target type if it is configured on the server.



Connection

Select the connection, FTP or SFTP server, where the data should be written.

Data File

Is the name of the data file pointed to by the synonym.

On IBM z/OS, to create a file in HFS (hierarchical file system), enter the name. To create a dataset, enter the name as

```
//'qualif.tablename.DATA'
```

where:

qualif

Is a fully qualified location.

tablename

Should match the synonym name.

Code Page

Specifies a code page for the target data.

Field Delimiter

Is the delimiter character(s) used between fields. The delimiter can be up to 30 characters. It can be specified as:

TAB. A tab character. This is the default.

a. A character string, for example ~.

0xnn. A hex code, for example, 0x44 (a comma), or 0x0D0A (a return and a linefeed). The hex code uses ASCII for Windows or UNIX systems and EBCDIC for IBM Mainframes.

Header

Inserts column headings as the first row of data and surrounds the column names with the character specified in the Enclosure field. When set to *Yes*, the FIELDNAME is used for the header.

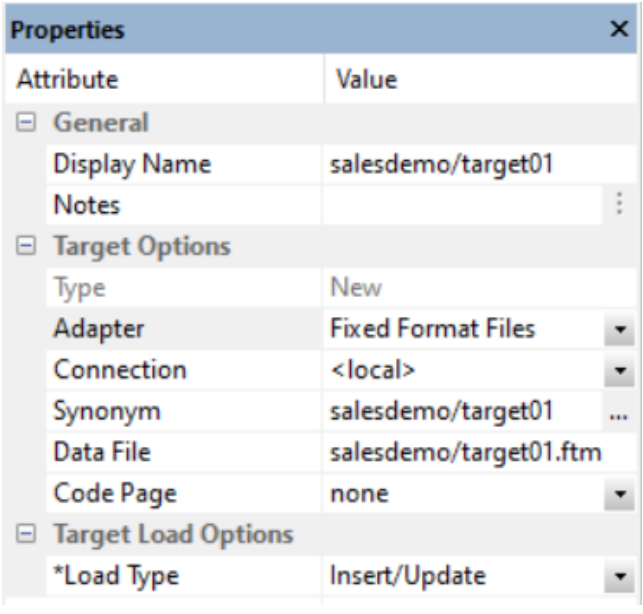
You can change this to use the TITLE as the header instead. To do so, expand the *Adapters* folder, right-click *Delimited Files (CSV/TAB)*, and click *Change Setting*s. From the HEADER drop-down menu, select *Build header based on field title* and click *Save*.

Enclosure

This character is used to surround the column headings when Header is set to *Yes*.

Reference: Target Properties Pane for Fixed Format File Targets

The Target Options section of the Properties pane for flat file targets contains the following additional options:



Connection

The connection configured for the selected adapter.

Data File

Is the name of the data file pointed to by the synonym.

On IBM z/OS, to create a file in HFS (hierarchical file system), enter the name. To create a dataset, enter the name as

```
//'qualif.tablename.DATA'
```

where:

qualif

Is a fully qualified location.

tablename

Should match the synonym name.

Code Page

Specifies a code page for the target data.

Prior to Load Options:

This option is only available for existing targets.

No changes

Does not delete the data target.

Delete File

Deletes and recreates the data target.

Reference: Target Properties Dialog Box for Formatted File Targets

The Target Options section of the Properties pane for formatted file targets contains the following additional options:

Format

Is the format of the target data.

Data File

Is the name of the data file pointed to by the synonym.

Procedure: How to Set Options for Fixed Format, Delimited, Formatted, and XML File Targets

1. In the data flow workspace, right-click the data target and click *Properties*.

   The Target Properties pane opens.

2. For existing targets, select whether to remove data prior to loading the data target in the *Prior to Load Options* section. The options are:

   No changes does not delete the data target. New records are appended.

   Delete File drops and recreates the data target.

3. Click *OK*.

   Note:

   - For fixed format files and XML files, the Load Type cannot be changed from Insert/Update, and the data is always appended.
   - For formatted file type EXL2K, the only Load Type available is Loading Fixed Format File using HOLD.

Loading Data to Salesforce.com

In order to insert or update rows in a Salesforce object, you can use a custom field identified as an External ID field.

Procedure: How to Create a Custom Field as an External ID

1. Log in to salesforce.com.
2. Click your name and select *Setup*.
3. Under App Setup, expand *Customize*. Select an object, for example, *Account*.
4. Click *Add a field to* on the object page.
5. Under Custom fields and relationships, click *New*.
6. Select a data type appropriate to an ID field, such as *Text* or *Number*. Click *OK*.
7. Enter a name, for example, *Account ID*. Fill in the other fields as desired.
8. Select the *External ID* checkbox.
9. Click *Next* twice and then click *Save*.

   The Account (or other object) Fields page opens. The new field is shown with its AP name, for example, Account_ID__C.
10. Create, or update, a synonym for Accounts (or another object).

Procedure: How to Identify an External ID Field In a Synonym

The synonyms created for Salesforce.com objects identify the ID field as the key. To use the External ID field as the key, you must edit the generated synonym and save a copy of it.

1. In the DMC, double-click the synonym for Account (or another object).
2. Right-click the *ACCOUNT* segment and click *Properties*.
3. For the attribute KEY, type the name of the External ID field, such as ACCOUNT_ID__C.
4. In the DMC, click *Save As*.
5. Enter a new name, such as accountx. Click *Save*.

Reference: Using an External ID Field to Insert or Update Rows in a Salesforce.com Object

Using the alternative synonym you can now use the External ID field to identify rows to be inserted or updated as you would with a table in a relational database. Salesforce.com will automatically do an *upsert*; if a value of the External ID field matches a value already in the object being loaded, an update will occur. If not, it will do an insert.

Reference: Deleting Rows from a Salesforce.com Object

The internal ID field is used to delete rows from a Salesforce.com object. To delete rows using the External ID, create a data flow that performs a DB_LOOKUP. You can also create a data flow that joins a table with a list of External ID values to be deleted to the Salesforce.com object to obtain the values of the corresponding ID field.

Mapping Transformations

After selecting a data source and target, you can create mappings to transform your data in the following ways:

- Map the columns of the source, or combinations of columns in the source, to columns in the data target.
- Convert the extract data into the form necessary for the data target.
- Create a temporary column for mapping.

  Note: You can create a temporary column within the mapping step to add intermediate calculations that are not part of the original SQL command or target data. This is often useful for complex calculations that are actually composed of multiple expressions. Creating a temporary column makes the transformation easier to understand and support. It is also helpful in performing data type conversions when multiple steps are required.

There are three methods for mapping columns:

- **Automapping.** Automatically maps columns with the same name.
- **Simple mapping.** Maps one column in the source to one in the target.
- **Complex mapping.** Builds expressions.

  Note: The target column format should be at least as large as the source column format, but you can adjust the transformation to accommodate different format sizes.

Reference: Considerations for Mapping to an Existing Fixed Format File

While you can map to an existing fixed format file, there are some considerations that should be taken into account.

- Fixed Format files do not have all of the options for loading data into the target that other data types do (for example, key matching logic).
- Data Migrator respects the data types described in the synonym of the fixed format file. For example, if the synonym describes numeric columns in integer format, data will be written to the file in that format.

- If a file already exists with the specified name, Data Migrator will append data to that file, unless you select the *Prior to Load* option of *Delete File*. Existing data in a fixed format file cannot be updated.

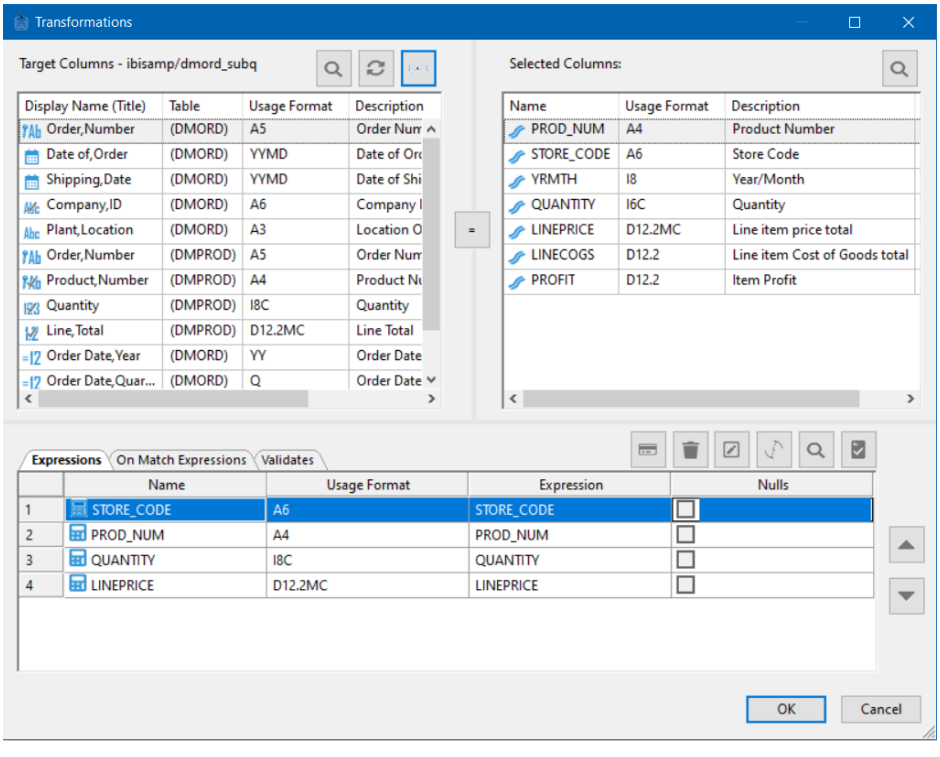
Procedure: How to Automatically Map All Columns

1. On the Home tab, in the Tools group, click *Options*.
2. Expand *Data Flow* and select *Data Flow Designer* from the User Preferences tree.
3. Select the *Automatically map all columns* check box in the On Add Target section.
4. Click *OK*.

Procedure: How to Map Transformations

1. From the data flow workspace, double-click a target, or right-click it and click *Target Transformations*.

   The Transformations dialog box opens, as shown in the following image.



2. To map columns with identical names, click the *AutoMap* button  .

The mapping will appear in the grid under the Expressions tab.

3. To map columns with different names, select one column from each grid and click the equal sign.

   The mapping will appear in the grid under the Expressions tab, with the Selected Column appearing in the Transform Expression box.

4. To create a complex mapping for a target column, double-click a column to move it to the Expressions tab.

5. Click the *Edit Transforms* button.

   The Transformation Calculator opens.

6. Create an expression using the Columns/Variables and Functions tabs, and the calculator buttons. For details on using the Transformation Calculator, see *Transformation Calculator* on page 199. For information about using functions, see *Using Functions* on page 325. For information about using variables, see *Using Variables in a Flow* on page 314.

   Note: When you create expressions, keep in mind their order of evaluation. For details, see *Order of Evaluation* on page 203

7. Click *OK*.

Note: Clicking a column heading will sort the column grid on that heading.

The new mapping will appear in the grid under the Expressions tab.

Note: The transformations will occur in the order in which they appear in the Expressions tab. If you need to change the order, select a transformation and use the up and down arrows.

Procedure: How to Create a Temporary Column for Mapping

1. In the data flow workspace, double-click a target, or right-click it and click *Target Transformations*.

   The Transformations dialog box opens.

2. Click the *Insert Transforms* button.

   The Transformation Calculator opens.

3. Type a name and type or select a format for the column.

4. Create an expression using the Columns/Variables and Functions tabs and the calculator buttons. For details on using the Transformation Calculator, see *Transformation Calculator* on page 199. For information about using functions, see *Using Functions* on page 325. For information about using variables, see *Using Variables in a Flow* on page 314.

   Note: When you create expressions, keep in mind their order of evaluation. For details, see *Order of Evaluation* on page 203.

5. Click *OK*.

The new mapping will appear in the grid under the Expressions tab.

The new mapping is now available as a column in the Transformation Calculator, and can be used to create a complex expression.

Tip: Since calculations are performed in the order they appear, make sure that your temporary column appears *before* any column in the Transform Rules box that uses it to calculate a value.

Procedure: How to Create a Target Transformation Using Relational Lookups

The DB_LOOKUP function allows you to build data flows that perform lookups into multiple tables using target transformations instead of explicit joins. This approach can simplify flows that require lookup values from numerous tables. For information about using the DB_LOOKUP function to create source transformations, see *Using Lookups to Create Multi-Table Flows Without Joins* on page 579.

1. From the data flow workspace, double-click a target object, or right-click it and click *Target Transformations*.

   The Transformations dialog box opens.

2. Double-click the column that will use the lookup. The column is moved to the grid under the Expressions tab.
3. With the column selected in the Expressions grid, click the *Edit Transforms* button.

   The Transformation Calculator opens.

4. Click the *Functions* tab, expand the *Data Source and Decoding - Simplified* folder, and double-click the *DB_LOOKUP* function.

   The prototype for DB_LOOKUP appears in the Expression window and the DB_LOOKUP dialog box opens.

5. Select a synonym to use for the lookup by clicking the ellipsis (...) after the *Lookup Synonym* field.

   The Select Synonym dialog box opens.

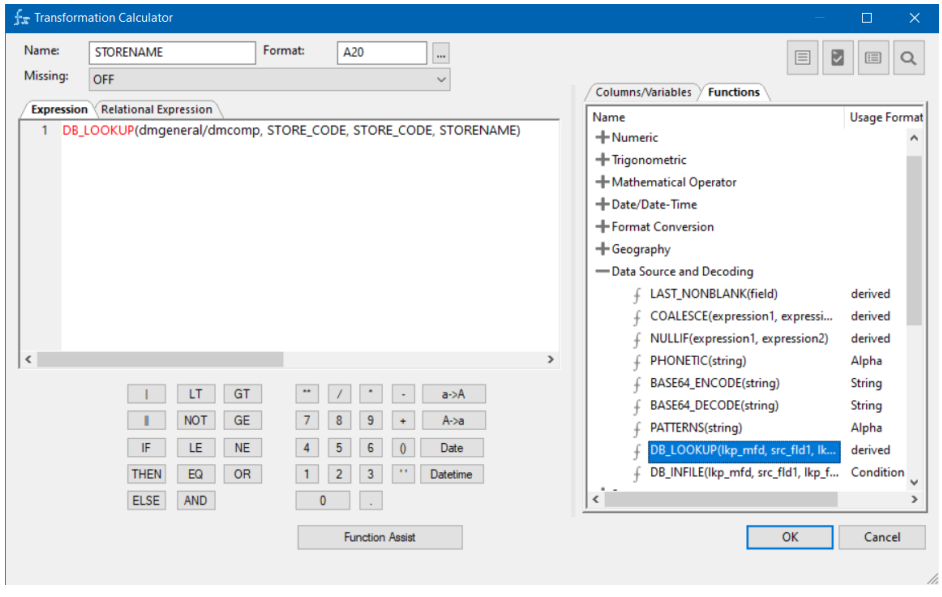
6. Select a synonym and click *Select*.
7. From the Lookup dialog box, click the *Add/Delete* button.
8. A lookup is similar to a join in that it uses columns from two different tables. Select a column from the Source and Lookup Fields lists, and click the *equal sign* (=).

Note: If you have *Automatically add join conditions* checked in the Data Management section of the Options dialog box, the Lookup is added automatically, since both data sources have a column with the same name and format, and that column is a key column in the lookup table.

9. Select the value to return from the lookup from the Return Field drop-down menu.
10. Select the original column name and click *OK*. Note that in this case, the format of the column is A20.
11. Click *OK*.

    The expression is completed in the Transformation Calculator dialog box.

    The Transformation Calculator should now look like this.



12. Click *OK*.

    The target transformation is added to the Transformations dialog box.

13. Click *OK* to close the Transformations dialog box.

Procedure: How to Edit a Transformation Mapping

From the Transformations dialog box:

1. Double-click the column number or name in the Expressions tab.

   or

Select the column name in the Expressions tab and click the *Edit transforms* button.

The Transformation Calculator opens.

2. Modify the expression.

   Note: You can use the *Function Assist* button to change the parameters of a transformation that uses the DB_LOOKUP function.

3. When you are finished with your changes click *OK*.

You can also double-click the *Expression* field in the Expressions tab, and enter the expression directly.

Procedure: How to Adjust a Transformation Mapping

If the format of the target column is smaller than the format of the source column, the transformation mapping will fail. For example, moving a column formatted as A24 into a column formatted as A10 will fail because you cannot move 24 characters into a 10-character field. To solve this problem, you would need to use edit or substring in a transform expression.

The Transformations dialog box provides an option for automatically adjusting the transformation to reconcile different format sizes. Adjust will also convert integer, alphanumeric, and datetime source columns so that they match the target column.

In the Transformations dialog box:

1. Map the source and target columns by selecting one from each list and clicking the equal sign.

   The mapping appears in the Expressions list.

2. With the expression highlighted, click the *Adjust Expression* button  button.

Procedure: How to Add Comments to a Transformation

1. Right-click the title bar under the Expression tab in the Transformations dialog box.
2. Point to *Customize* from the shortcut menu and then click *Description*.

The Transformations dialog box now shows a Description column. Here you can enter a comment for the transformation that is stored in the flow.

Procedure: How to Test Mappings

From the Transformations dialog box:

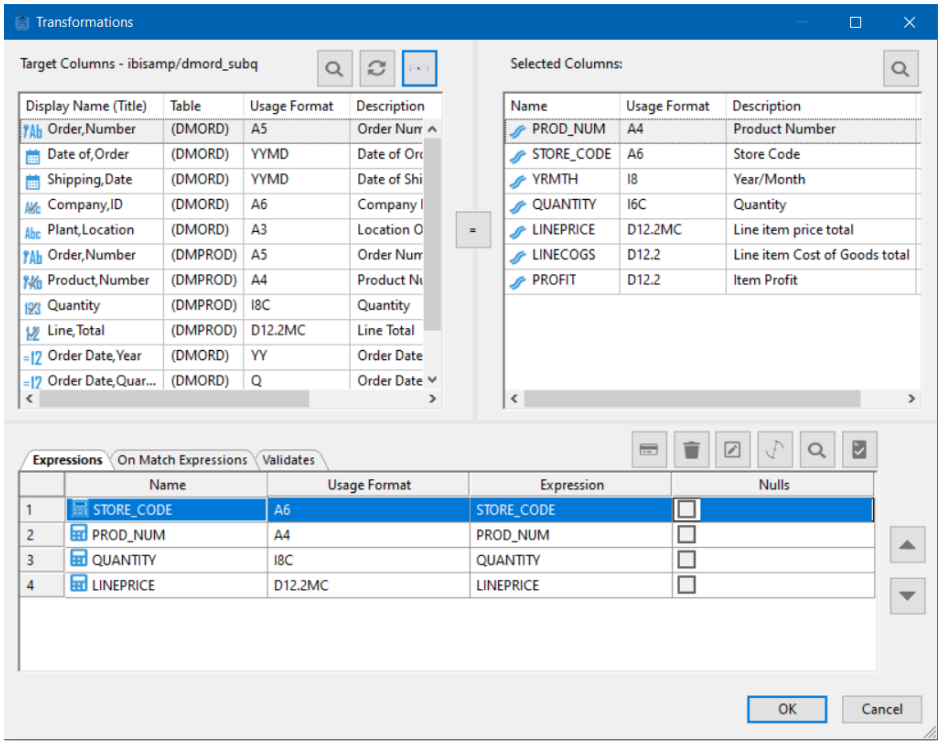
1. Click the *Test Transforms*  button.

The Test Transformations dialog box opens with sample data for the transformations.

2. Click the *Close* button.

Reference: Transformations Dialog Box

To access the Transformations dialog box from the data flow workspace, double-click a target, or right-click a target and click *Target Transformations*.



The Transformations dialog box contains the following fields and options:

Find

Allows you to search for a column.

Refresh

Refreshes the columns in the target.

Automap

Maps all columns with identical names.

Target Columns

Lists the columns in the data target.

Note: Clicking a column heading will sort the column grid on that heading.

Selected Columns

Lists the columns in the data source.

=

Establishes the relationship between the selected columns as equal.

 **Insert Transforms**

Inserts a new mapping and opens the Transformation Calculator.

 **Delete**

Deletes a selected mapping.

 **Edit Transforms**

Opens the Transformation Calculator to edit the selected mapping.

 **Adjust Expressions**

Adjusts the transformation expression to reconcile different format sizes.

Expressions tab

Lists the transformation mappings. To access the context menu with available options for the transformations, right-click a transformation in this tab. For more information, see *Source Transformations Shortcut Menu* on page 198.

On Match Expressions tab

Lists the expressions that are performed for rows that are updated. This tab is not available if *Optimize Load* is enabled in the flow properties.

Validates tab

Lists transformations that screen records as they are loaded into the data target. If a record does not meet the criteria, it is rejected. This tab is not available if *Optimize Load* is enabled in the flow properties.

Test transforms

Retrieves, by default, 50 records and 4096 characters per row from the source, and applies all available transformations. This option enables you to review the data being moved and to ensure that the syntax of each transformation is correct. No duplicate processing is performed. For example, if a key value is duplicated, a unique constraint violation may occur. Similarly, if a value is missing for a column described as not null, a constraint violation will occur at run time.

Note: You can set the number of rows retrieved from the Options dialog box under the *Run Options* branch of the *User Preferences* tree. The default is 50 rows.

If there are any errors, the Server Messages window displays an error message and marks the incorrect transformation in red.

Up and Down arrows

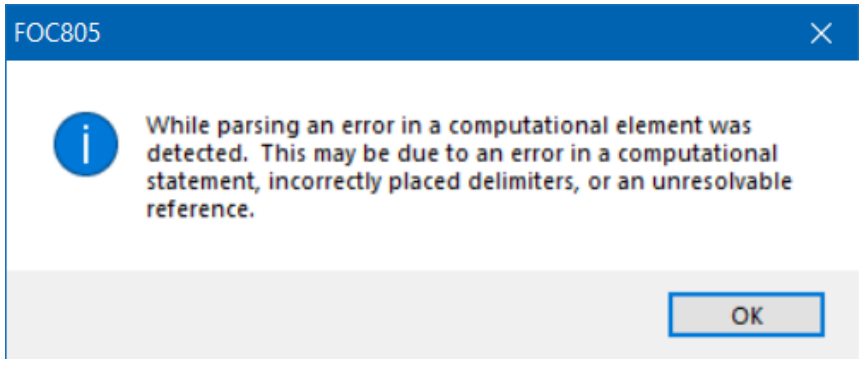
Move the selected mapping to change the order of the transformations.

Procedure: How to View Error Details

Error details for transformations can be viewed in the DMC. If an error is made, the transform will appear in red.

1. Right-click the error and select *Error Details*.
2. The Error(s) Details dialog box opens.

   To see the error message details, click the error message number. A window opens explaining the error, as seen in the following image.



3. Click *OK*.

   Note: This method of viewing error details also applies to target transformations.

Validating Records

You can create a business rule that screens records as they are loaded into the data target. If a record does not meet the criteria, it is rejected.

If a value meets the criteria of the rule, a value of 1 is assigned to the record and the record is loaded. If the value does not meet the criteria of the record, the value of 0 is assigned to the record and the record is rejected.

Procedure: How to Validate Records

1. In the Transformations dialog box, select the *Validates* tab.
2. Click the *Insert Intermediate Transforms* button  .

   The Transformation Calculator opens with a name and format already provided.
3. Optionally, enter a name for the validation rule in the Column field.
4. Create an expression for the validation rule using the Columns, Functions, and Variables tabs, and the calculator buttons.

   If the expression evaluates to TRUE then the row is loaded in the target table. If the expression evaluates to FALSE then the row is rejected as invalid.

   Note that the validation is performed after any filters or aggregation. For example, if you enter QUANTITY GT 10, only records where the sum of the values of QUANTITY for the group by fields is greater than 10 will be loaded.

   The count of rejected rows is reported in the Data Migrator log as *invalid* records. These records can also be logged to a file for review. For more information, see *Transformation Calculator* on page 199.
5. Click *OK*.

The validation rule appears in the grid below the Validates tab.

Using Variables in a Flow

You can use a variable in your data flow to represent a value in a transformation. For process flows, variables are typically used in stored procedures or to control the flow.

The following types of variables can be used in a flow.

- Data Migrator variables are supplied with the DMC.
- System variables are supplied with the DMC.

- User-defined variables are created by a user.

  Note: You can set a value for a variable in the set variables objects on the process flow tab. You can also specify run-time parameters directly from a batch file, shell script, or JCL. For information on setting values when executing flows from CMASAP or CMRUN, see *Event-Based Scheduling* on page 681.

- Statistical and Control variables can be referenced in stored procedures. They are primarily used in process flows.

Reference: Data Migrator Variables

Data Migrator variables are available from the Variables tab in the calculators. Data Migrator variables start with &&CM__ (double underscores). They include:

&&CM__AUTHOR

Is the user ID that originally saved the flow.

Note: &&CM__AUTHOR is the user ID that originally saved the flow, which is not always the user ID that ran the flow. To find the user ID that ran the flow, assign a value of GETUSER('A32') to a variable in the set variables object on the Process Flow tab.

&&CM__REQUEST

Is the name of the flow.

&&CM__TARGET

Is the name of the target table for the data flow. Note that for flows with multiple target tables, the variable CM__TARGET has a number appended to the name.

Reference: Data Migrator Return Code Variable

You can test for the return code from a flow, or pass the return code to another variable.

&&CM__RETURN

Is the return code from a data flow. The value zero (0) means the flow processed without exceeding the specified number of DBMS errors (10000000, by default). There still may be FOCEXEC errors, which you can check using the &FOCERRNUM variable. If you want processing to stop when &&CM__RETURN is non-zero, open the Flow Properties and, in the Execution section, enter the number of DBMS errors you want to allow in the *Stop after DBMS errors* text box.

Reference: Control Variables

Control variables are typically used in process flows.

&&KILL_RPC

Shuts down all flow processing if the value is set to Y. The flow immediately branches to the end of its logic, skipping all code that follows the stored procedure where this setting is used. Any data flows or procedures are not executed. This may be useful in cases where a site-dependent variable (program result, special file allocation, and so on) fails, and the entire extract is to be stopped. The default is N.

To reset, assign a value of 'Y" to the global variable KILL_RPC in a set variables object on the process flow tab.

Reference: System Variables

System variables are identified by a single ampersand and an alphanumeric name, up to ten characters long. These variables can be referenced in stored procedures. Each date shown below represents May 25, 2005.

&DATE

Is the current date (MM/DD/YY) as a character string (A8).

Example: 05/25/05

&DATEDMY

Is the current date (DD/MM/YY) as a character string (A8).

Example: 25/05/05

&DATEDMYY

Is the current four-digit year date (DD/MM/CCYY) as a character string (A10).

Example: 25/05/2005

&DATEMDY

Is the current date (MM/DD/YY) as a character string (A8).

Example: 05/25/05

&DATEMDYY

Is the current four-digit year date (MM/DD/CCYY) as a character string (A10).

Example: 05/25/2005

&DATEYMD

Is the current date (YY/MM/DD) as a character string (A8).

Example: 05/05/25

&DATEYYMD

Is the current four-digit year date (CCYY/MM/DD) as a character string (A10).

Example: 2005/05/25

&DMY

Is the current date as day, month, year. Useful for numerical comparisons.

Example: 250505

&DMYY

Is the current date as day, month, with a four-digit year.

Example: 25052005

&MDY

Is the current date as month, day, year.

Example: 052505

&MDYY

Is the current date as month, day, with a four-digit year.

Example: 05252005

&TOD

Is the current time. (HH.MM.SS)

Example: 17:00:00

&YMD

Is the current date as year, month, day.

Example: 050525

&YYMD

Is the current date as a four-digit year, month, day.

Example: 20050525

Reference: Statistical Variables

Data Migrator supports the following statistical variables that can be referenced in stored procedures. Statistical variables are recorded up to a value of 2,147,483,647. They are primarily used in process flows.

&ACCEPTS

Is the number of transactions accepted.

&CHNGD

Is the number of rows updated.

&DBMSERR

Is the number of rows rejected due to database constraint violations, or for fixed format file targets due to a format error.

&DELTD

Is the number of rows deleted.

&DUPLS

Is the number of rows rejected as a result of duplicate values in the data source.

&EXITRC

Is return code value from the execution of an operating system command.

&FORMAT

Is the number of rows rejected as a result of a format error, or when a field is truncated.

&INPUT

Is the number of rows added to the data source.

&INVALID

Is the number of rows rejected as a result of an invalid condition.

&NOMATCH

Is the number of rows rejected as a result of not matching a value in the data source.

&REJECTS

Is the number of rows rejected for reasons other than the ones specifically tracked by other statistical variables.

&TRANS

Is the number of rows processed.

&LINES

Is the number lines returned in the last answer set.

&RECORDS

Is the number of records retrieved in last answer set.

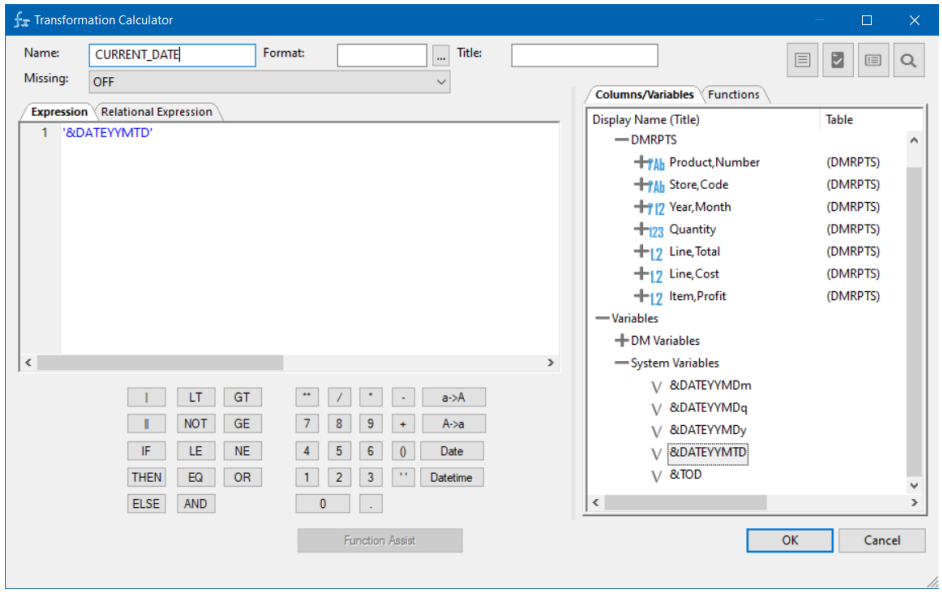
Procedure: How to Use Data Migrator or System Variables

1. From a calculator, select the *Columns/Variables* tab.

   Note: The variables are grouped by category. To view all variables, right-click in the variables workspace and click *View Alphabetical*.

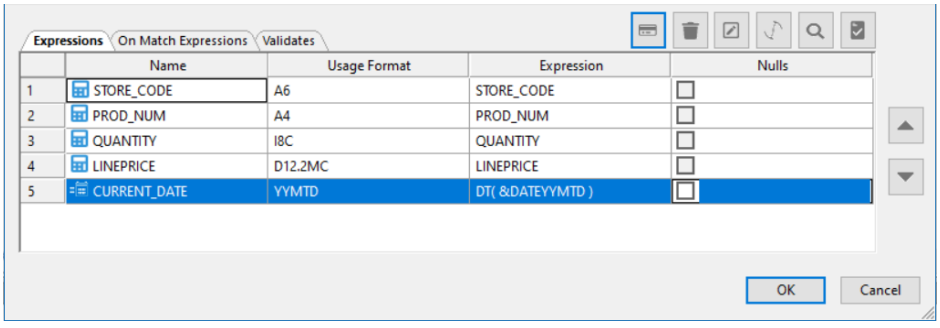
2. Double-click a variable to add it to an expression or condition. You can also type in the name of the variable. Data Migrator and System variables must be enclosed in single quotation marks (‘).

   Note: Match the format (alphanumeric or numeric) to the system variable in a Source or Target Transformation. &TOD and system variables that start with &DATE are alphanumeric. All others are numeric.



3. When the expression is complete, click *OK*.

The variable appears in the Expression column in the Transformations window, as shown in the following page.



The Variables tab is also available in the SQL and Filter calculators.

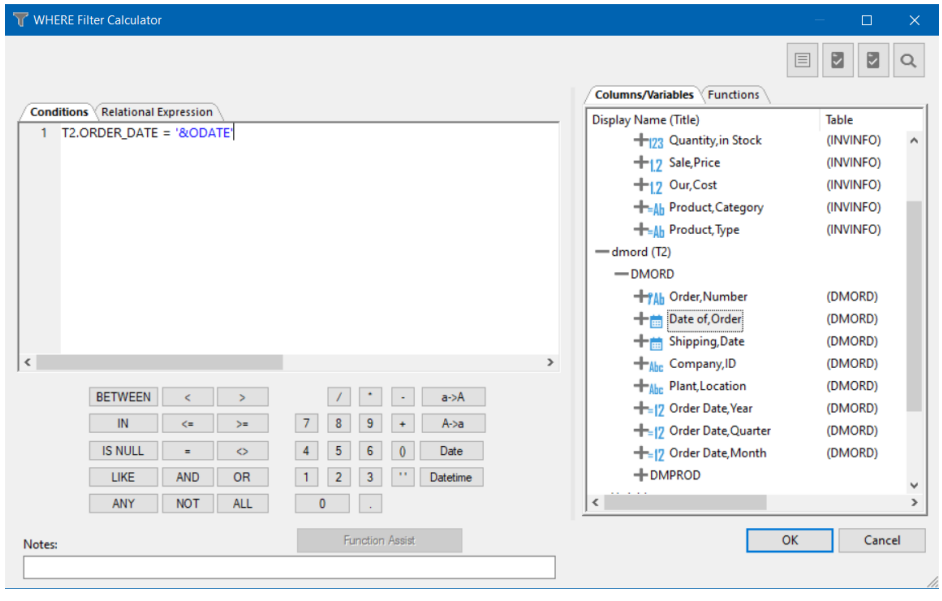
Using a Parameter With a Flow

Variables that are used as parameters are called local variables and are identified within a flow by a single ampersand and an alphanumeric name, up to ten characters long. Local variables can be used throughout the flow, for example, in transformations or filters. In order to use a local variable it must have a value.

- **Establishing local variables when designing a flow.** If a flow contains a local variable and you run any test, you will be prompted to enter a value for the variable.
- **Setting a value when running a flow.** Local variables are set in a process flow that calls a data flow in the flow properties.

Procedure: How to Use a Local Variable as a Passed Parameter

1. You can use a local variable, a variable known only to a particular procedure, as a passed parameter.
2. Type the variable name into an expression or condition in one of the calculators. The variable name must be enclosed in single quotes if it contains alphanumeric or data values, as shown in the following image.

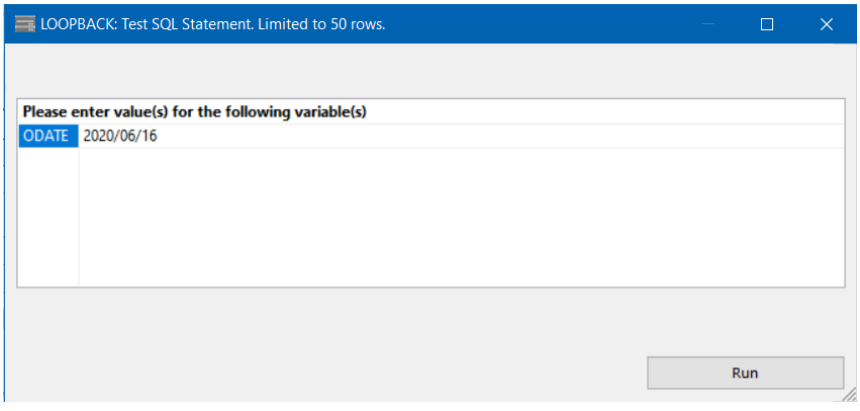


3. Click *OK*.

Procedure: How to Test a Local Variable

1. Open the expression in a calculator.
2. Click the *Test SQL Statement* button.
3. Data Migrator prompts you to enter a value for the variable.

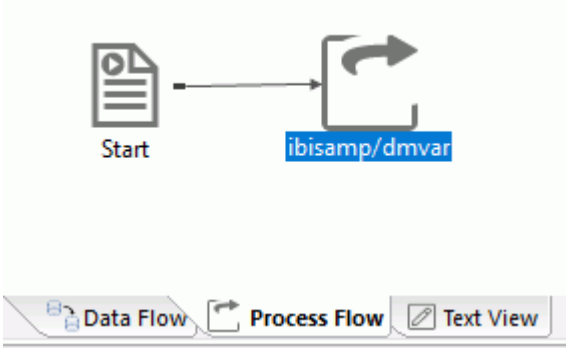
Note that when entering a date value, no quotes are required, and the date must be entered in the format used in a where condition, as shown in the following image.



4. Click *Run*.

Procedure: How to Pass a Parameter Using a Local Variable

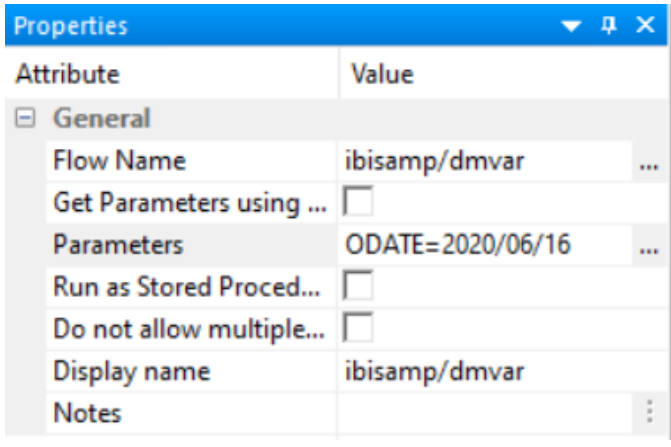
1. Create a process flow.
2. In the Process Flow tab, drag the data flow that uses the local variable into the workspace, to the right of the Start object.



3. Double-click the Data Flow object (ibisamp/dmvar), or right-click the object and click *Properties*.

   The Properties panel opens.

4. Enter the value for the parameter in the Parameters field, as shown in the following image.



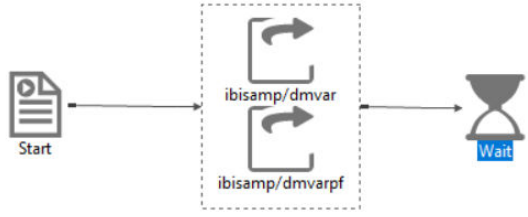
Note that you do not need to manually enter the parameter name and value. Clicking the ellipsis button to the right will bring up a dialog box where you can enter values for the parameters. If you choose to enter a date value manually, no quotes are required, and the date must be entered in the format used in a where condition. Multiple parameters can be separated by commas. Also note that checking *Run as Stored Procedure* is not required.

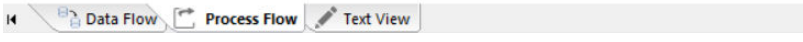
5. Click *OK*.
6. Save and run the flow.

Procedure: How to Pass a Local Variable as a Parameter in a Parallel Group

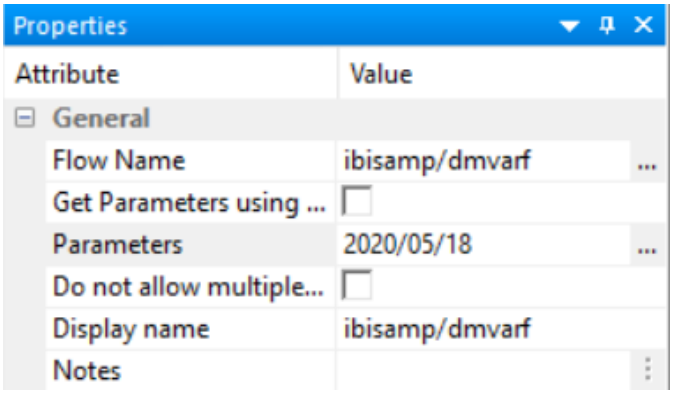
1. Create a process flow.
2. Add a Parallel Group.

3. Drag the Data Flows that you want to run inside the Parallel Group.



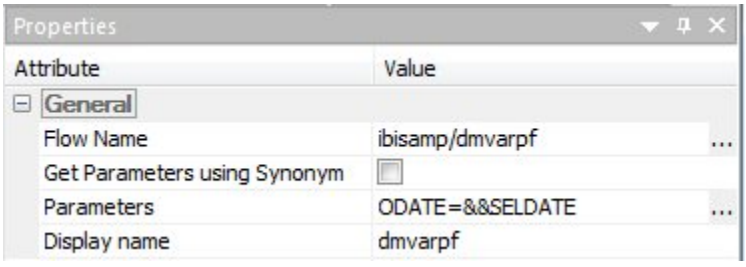


4. Double-click the data flow that requires a parameter, or right-click it and click *Properties*.
5. Enter the value for the parameter in the Parameters field, as shown in the following image.



Note that you do not need to manually enter the parameter name and value. Clicking the ellipsis button to the right will bring up a dialog box where you can enter values for the parameters. If you choose to enter a date value manually, no quotes are required and the date must be entered in the format used in a where condition. Multiple parameters can be separated by commas.

6. Alternatively, the value for the local variable can be a (global) variable set in a stored procedure, as shown in the following image.



7. Click *OK*.
8. Save and run the flow.

Using Functions

When you click the *Functions* tab in the Transformation, Filter, or SQL calculators, a list of available functions appears. Right-click the Functions tab to display them alphabetically or by category. For a list of available SQL functions, see *SQL Functions* on page 231.

When using functions in a calculator, the Function Assist dialog box can help you set the parameters.

Note: All functions may not be available on all platforms.

Function	Arguments	Description
Character Functions		
ARGLEN	(*length, source_string, output_format*)	Measures the length of a character string within a field, excluding trailing blanks.
ASIS	(*argument*)	Distinguishes between a space and a zero. **Note:** This function is only available for variables and appears only in the User Variables calculator.

Function	Arguments	Description
CHKFMT	(*numchar*, *source_string*, '*mask*', *output_format*)	Checks numeric and alphanumeric fields for invalid character types.
CTRAN	(*length*, *source_string*, *decimal*, *decvalue*, *output_format*)	Replaces characters in a string, taking decimal ASCII values as arguments for the target.
CTRFLD	(*source_string*, *length*, *output_format*)	Centers a character string within a field.
EDIT	(*source_string*)	Converts between alphanumeric and integer data.
EDIT	(*source_string*, '*mask*')	Using a mask, extracts characters from an alphanumeric string and inserts new characters.
GETTOK	(*source_string*, *length*, *token_number*, '*delim*', *outlen*, *output_format*)	Divides a string into tokens and returns the specified token.
LCWORD	(*length*, *source_string*, *output_format*)	Converts the letters in a given string to mixed-case: the first letter of each new word and the first letter after a single or double quotation mark are converted to uppercase. The rest are converted to lowercase.
LCWORD2	(*length*, *source_string*, *output_format*)	Converts a character string to mixed-case. The first letter of every word is converted to uppercase.

Function	Arguments	Description
LCWORD3	(*length*, *source_string*, *output_format*)	Converts a character string to mixed-case; the first letter after every apostrophe remains capitalized and every other letter is converted to lowercase, unless it is the last letter of the word. If it is the last letter, it will be converted to lowercase.
LJUST	(*length*, *source_string*, *output_format*)	Left-justifies a character string within its field.
LOCASE	(*length*, *source_string*, *output_format*)	Returns the infield in lowercase.
OVRLAY	(*source_string*, *length*, *substring*, *sublen*, *position*, *output_format*)	Overlays a substring to a specified place on another character string.
PARAG	(*length*, *source_string*, *'delim'*, *max_token_size*, *output_format*)	Divides lines of text into smaller lines by marking them off with a delimiter character.
PATTERN	(*length*, *source_string*, *output_format*)	Converts text to its pattern.
PATTERNS	(*string*)	Returns a character string that represents the pattern of the input string. Letters appear as 'A' or 'a', digits appear as '9', and special characters appear as is. This function is available starting in release 7708.

Function	Arguments	Description
POSIT	(*source_string*, *length*, *substring*, *sublength*, *output_format*)	Finds the starting position of a substring within a larger string.
REPLACE	(*input_string*, *search_string*, *replacement*)	Replaces all instances of a search string in an input string with the given replacement string.
RJUST	(*length*, *source_string*, *output_format*)	Right-justifies a character string within a field.
SOUNDEX	(*length*, *source_string*, *output_format*)	Converts character strings to 4-letter codes, enabling you to search for character strings phonetically without knowing how they are spelled.
SPELLNM	(*outlength*, *number*, *output_format*)	Takes a number with two decimal spaces and spells it out with dollars and cents.
SPLIT	(*element*, *string*)	Returns a specific type of element from a string. The output is returned as variable length alphanumeric.
SQUEEZ	(*length*, *source_string*, *output_format*)	Reduces multiple contiguous blank characters within an input string to a single blank character.
STRIP	(*length*, *source_string*, *char*, *output_format*)	Removes all occurrences of a specific character from an input string.

Function	Arguments	Description
STRREP	(*inlength, instring, searchlength, searchstring, replength, repstring, outlength, output_format*)	Replaces character strings.
SUBSTR	(*length, source_string, start, end, sublength, output_format*)	Extracts a substring of a given length and starting position from a larger string.
TRIM	(*trim_where, source_string, length, pattern, sublength, output_format*)	Removes leading and/or trailing occurrences of a pattern within a string.
TRUNCATE	(*ARGUMENT*)	Removes trailing blanks from Dialogue Manager amper variables, and adjusts the length accordingly. **Note:** This function is only available for variables and appears only in the User Variables calculator.
UPCASE	(*length, source_string, output_format*)	Converts all characters in a string to uppercase.
Character Functions - DBCS Code Pages		
DCTRAN	(*length, source_string, decimal, decvalue, output_format*)	Translates a single-byte or double-byte character within a character string to another character based on its decimal value.

Function	Arguments	Description
DEDIT	(*length*, *source_string*, *mask_length*, *'mask'*, *output_format*)	Extracts characters from or adds characters to a string when server is configured to use a DBCS code page.
DPART	(*date*, *'component'*, *output_format*)	Retrieves a date component as a numeric value.
DSTRIP	(*length*, *source_string*, *char*, *output_format*)	Removes all occurrences of a specific single-byte or double-byte character from a string.
DSUBSTR	(*length*, *source_string*, *start*, *end*, *sublength*, *output_format*)	Extracts a substring based on its length and position in the parent string when server is configured to use a DBCS code page.
Character Functions - Variable Length		
LENV	(*source_string*, *output_format*)	Returns the actual length of an A*n*V input field or the size of an A*n* field.
LOCASV	(*upper_limit*, *source_string*, *output_format*)	Converts alphabetic characters to lowercase and is similar to LOCASE.
POSITV	(*source_string*, *upper_limit*, *substring*, *sub_limit*, *output_format*)	Finds the starting position of a substring within a larger string and is similar to POSIT.
SUBSTV	(*upper_limit*, *source_string*, *start*, *sub_limit*, *output_format*)	Extracts a substring from a string and is similar to SUBSTR.

Function	Arguments	Description
TRIMV	(*trim_where*, *source_string*, *upper_limit*, *pattern*, *pattern_limit*, *output_format*)	Removes a pattern from a string and is similar to TRIM.
UPCASV	(*upper_limit*, *source_string*, *output_format*)	Converts alphabetic characters to uppercase and is similar to UPCASE.
Data Source and Decoding Functions		
COALESCE	(*expression1*, *expression2* [,...])	Evaluates the arguments in order, and returns the first argument that is not MISSING.
DB_LOOKUP	(*lookupsynonym*, *lookupcolumn1*, *sourcecolumn1*, *lookupcolumn2*, *sourcecolumn2* ... *returncolumn*)	Looks up values in a table and returns the value from another column of the row where the first value was found. For more information on improving lookup performance, see *Improving Lookup Performance* on page 480.
DB_EXPR	(*native_SQL_expression*)	Inserts a native SQL expression exactly as entered into the native SQL generated for a FOCUS or SQL language request.
DB_INFILE	(*target_file*, *s1*, *t1*, ... *sn*, *tn*)	Compares one or more field values in a source file to values in a target file.
DECODE	*fieldname*(*code1* *'result1'*... [*ELSE* *default*])	Changes a variable to an associated name.
LAST	*fieldname*	Retrieves the previous value for a field.

Function	Arguments	Description
NULLIF	(*expression1*, *expression2*)	Returns MISSING if the value of the two arguments are equal. Otherwise it returns the first value.
Date Functions - Legacy		
AYM	(*indate*, *months*, *output_format*)	Adds or subtracts months from dates stored in alphanumeric or integer format.
AYMD	(*indate*, *days*, *output_format*)	Adds or subtracts a given number of days to a date in [YY]YYMMDD format.
CHGDAT	('*in_display_options*', '*out_display_options*', *date_string*, *output_format*)	Rearranges the year, month, and day portions of an input character string representing a date. It may also convert the input string from long to short or short to long date representation.
DADMY	(*indate*, *output_format*)	Given the date in day-month-year format, calculates the number of days since 1/1/1900.
DADYM	(*indate*, *output_format*)	Calculates the number of days since 1/1/1900 given the date in day-year-month format.
DAMDY	(*indate*, *output_format*)	Calculates the number of days since 1/1/1900 given the date in month-day-year format.
DAMYD	(*indate*, *output_format*)	Calculates the number of days since 1/1/1900 given the date in month-year-day format.
DAYDM	(*indate*, *output_format*)	Calculates the number of days since 1/1/1900 given the date in year-day-month format.

Function	Arguments	Description
DAYMD	(*indate*, *output_format*)	Calculates the number of days since 1/1/1900 given the date in year-month-day format.
DMY	(*from_date*, *to_date*)	Returns the difference between two dates in day-month-year order.
DOWK	(*indate*, *output_format*)	Returns the day of the week in A4 format.
DOWKL	(*indate*, *output_format*)	Returns the day of the week in A12 format.
DTDMY	(*number*, *output_format*)	Returns date in day-month-year format given the number of days since 1/1/1900.
DTDYM	(*number*, *output_format*)	Given the number of days since 1/1/1900, returns date in day-year-month format.
DTMDY	(*number*, *output_format*)	Returns date in month-day-year format given the number of days since 1/1/1900.
DTMYD	(*number*, *output_format*)	Returns date in month-year-day format given the number of days since 1/1/1900.
DTYDM	(*number*, *output_format*)	Returns date in year-day-month format given the number of days since 1/1/1900.
DTYMD	(*number*, *output_format*)	Returns date in year-month-day format given the number of days since 1/1/1900.
GREGDT	(*indate*, *output_format*)	Converts a Julian date to a Gregorian date.

Function	Arguments	Description
JULDAT	(*indate*, *output_format*)	Converts a Gregorian date to a Julian date.
MDY	(*from_date*, *to_date*)	Returns the difference between two dates in month-day-year order.
TODAY	(*output_format*)	Returns the current date in alphanumeric format (MM/DD/YYYY).
YM	(*from_date*, *to_date*, *output_format*)	Returns the number of days between two dates.
YMD	(*from_date*, *to_date*)	Returns the difference between two dates in year-month-day order.
Date Functions - Standard		
DATEADD	(*date*, *component_code*, *increment*)	Adds or subtracts years, months, or days to or from a date.
DATECVT	(*date*, *in_format*, *out_format*)	Converts dates from one date format to another.
DATEDIF	(*from_date*, *to_date*, *component_code*)	Calculates the difference between two dates, expressed as years, months, or days.
DATEMOV	(*date*, '*move-point*')	Moves a date to a significant point on the calendar.
DATETRAN	(*indate*, '(*intype*)', '(*formatops*)', '*lang*', *outlen*, *output*)	Formats dates in international formats.
Date-Time Functions		

Function	Arguments	Description
DTADD	(*date*, *component*, *increment*)	Adds or subtracts the specified number of calendar or time components to the input date or date-time fields, and returns the new date or date-time value.
DTDIFF	(*end_date*, *start_date*, *component*)	Given two dates in standard date or date-time formats, returns the number of a given component boundaries between the two dates.
HADD	(*timestamp*, *component*, *increment*, *length*, *output_format*)	Increments a date-time field by a given number of units.
HCNVRT	(*timestamp*, (*format*), *length*, *output_format*)	Converts a date-time field to alphanumeric format for use with operators, such as EDIT, CONTAINS, and LIKE.
HDATE	(*timestamp*, *output_format*)	Extracts the date portion of a date-time field and converts it to a date format.
HDIFF	(*end_timestamp*, *start_timestamp*, *component*, *output_format*)	Finds the number of boundaries of a given type crossed going from date 2 to date 1.
HDTTM	(*date*, *length*, *output_format*)	Converts a date field to a date-time field. The time portion is set to midnight.
HEXTR	(*source*, *'componentstring'*, *length*, *output_format*)	Extracts components of a date-time value and sets remaining components to zero.
HGETC	(*length*, *output_format*)	Stores the current date and time in a date-time field.

Function	Arguments	Description
HHMMSS	(*output_format*)	Returns the current time in alphanumeric format.
HINPUT	(*source_length*, *source_string*, *timestamp_length*, *output_format*)	Converts an alphanumeric string to a date-time value.
HMASK	(*source*, *'componentstring'*, *input*, *length*, *output_format*)	Extracts components of a date-time value and preserves remaining components.
HMIDNT	(*timestamp*, *length*, *output_format*)	Changes the time portion of a date-time field to midnight (all zeroes).
HNAME	(*timestamp*, *component*, *output_format*)	Extracts a specified component from a date-time field and returns it in alphanumeric format.
HPART	(*value*, *component*, *output_format*)	Extracts a specified component from a date-time field and returns it in numeric format.
HSETPT	(*timestamp*, *component*, *value*, *length*, *output_format*)	Inserts the numeric value of a specified component into a date-time field.
HTIME	(*length*, *timestamp*, *output_format*)	Converts the time portion of a date-time field to a numeric number of milliseconds (if the first argument is 8) or microseconds (if the first argument is 10).
HTMTOTS	(*time*, *length*, *output_format*)	Converts a time to a timestamp.

Function	Arguments	Description
Format Conversion Functions		
EDIT	(*fieldname*)	Converts between alphanumeric and integer data. Used with a mask, extracts characters from an alphanumeric string and inserts new characters.
FTOA	(*number*, '(*format*)', *output_format*)	Converts numbers from decimal format to alphanumeric format.
ITONUM	(*maxbytes*, *infield*, *output_format*)	Converts large binary integers in non-FOCUS files to double-precision format.
ITOPACK	(*maxbytes*, *infield*, *output_format*)	Converts large binary integers in non-FOCUS files to packed-decimal format.
ITOZ	(*length*, *in_value*, *output_format*)	Converts numbers from numeric format to zoned format for extract files.
PTOA	(*number*, '(*format*)', *output_format*)	Converts a number from numeric format to alphanumeric format.
TO_INTEGER	(*string*)	Converts a character string that contains a valid number consisting of digits and an optional decimal point to an integer value.
TO_NUMBER	(*string*)	Converts a character string that contains a valid number consisting of digits and an optional decimal point to the numeric format most appropriate to the context.

Function	Arguments	Description
TSTOPACK	(*timestamp*, *'P2'*)	Converts an MS SQL Server or Sybase column described as "timestamp", which contains an increasing counter that represents a unique value in the database in hex notation with a format of A16 into a packed decimal number with a format of P21.
Legacy Functions		
ASIS	(*argument*)	Distinguishes between a space and a zero.
ATODBL	(*source_string*, *length*, *output_format*)	Converts a number in alphanumeric format to decimal format.
BAR	(*barlength*, *infield*, *maxvalue*, *'char'*, *output_format*)	Produces a bar chart.
BITSON	(*bitnumber*, *source_string*, *output_format*)	Evaluates an individual bit within a character string to determine whether it is on or off.
BITVAL	(*source_string*, *startbit*, *number*, *output_format*)	Evaluates a string of bits within character strings and returns its binary value.
BYTVAL	(*character*, *output_format*)	Translates a character to its corresponding ASCII code.
CHKPCK	(*length*, *in_value*, *error*, *output_format*)	Validates the data in a field described as packed format.

Function	Arguments	Description
FIND	(*fieldname* [AS *dbfield*] IN *file*);	Verifies if a value exists in an indexed field in another file.
HEXBYT	(*decimal_value*, *output*)	Converts a numeric value to its corresponding ASCII character.
LOOKUP	*field*	Retrieves a data value from a cross-referenced FOCUS data source in a MODIFY request.
PCKOUT	(*in_value*, *length*, *output_format*)	Writes packed numbers of varying lengths (between one and 16 bytes) to extract files.
REVERSE	(*length*, *source_string*, *output_format*)	Reverses the input characters.
UFMT	(*source_string*, *length*, *output_format*)	Converts input field values to hexadecimal (hex) notation.
Numeric Functions		
ABS	(*in_value*)	Returns the absolute value of its argument.
CEIL	(*number*)	Returns the smallest integer that is greater than or equal to the number.
DMOD	(*dividend*, *divisor*, *output_format*)	Returns the remainder from a division as a number in decimal format.
EXP	(*power*, *output_format*)	Raises the number e to a power you specify.

Function	Arguments	Description
EXPN	(*n.nn* {E|D} {+|-} *p*)	Evaluates an argument expressed in scientific notation.
FLOOR	(*number*)	Returns the largest integer that is less than or equal to the number.
FMOD	(*dividend*, *divisor*, *output_format*)	Returns the remainder from a division as a number in floating-point format.
IMOD	(*dividend*, *divisor*, *output_format*)	Returns the remainder from a division as a number in integer format.
INT	(*in_value*)	Returns the integer part of its argument.
LOG	(*in_value*)	Returns the natural logarithm of its argument.
MAX	(*value1*, *value2*, ...)	Returns the maximum value from its list of arguments.
MIN	(*value1*, *value2*, ...)	Returns the minimum value from its list of arguments.
MOD	(*dividend*, *divisor*)	Calculates the remainder from a division as an integer.
NORMSDST	(*value*, *output_format*)	Calculates the cumulative normal standard distribution.
NORMSINV	(*value*, *output_format*)	Calculates the inverse cumulative normal standard distribution.
PRDNOR	(*seed*, *output_format*)	Generates reproducible random numbers.
PRDUNI	(*seed*, *output_format*)	Generates reproducible random numbers.

Function	Arguments	Description
RDNORM	(*output_format*)	Generates random numbers.
RDUNIF	(*output_format*)	Generates random numbers.
SQRT	(*in_value*)	Returns the square root of its argument.
System Functions		
APINPATH	(*length*, *appname*, 'A1')	Returns Y if application appname is in the path, N if not.
CLSDDREC	(*output_format*)	Closes all files opened by the PUTDDREC function.
FEXERR	(*error*, 'A72')	Returns the first line of the error message.
FGETENV	(*length*, *varname*, *outlen*, *output_format*)	Gets the value of an environmental variable and returns it as an alphanumeric string.
FPUTENV	(*varname_length*, *varname*, *value_length*, *value*, *output_format*)	Assigns a character string to an environmental variable.
GETUSER	(*output_format*)	Retrieves the user ID from the system.
PUTDDREC	(*ddname*, *dd_len*, *record_string*, *record_len*, *output_format*)	Writes a character string as a record in a fixed format file.

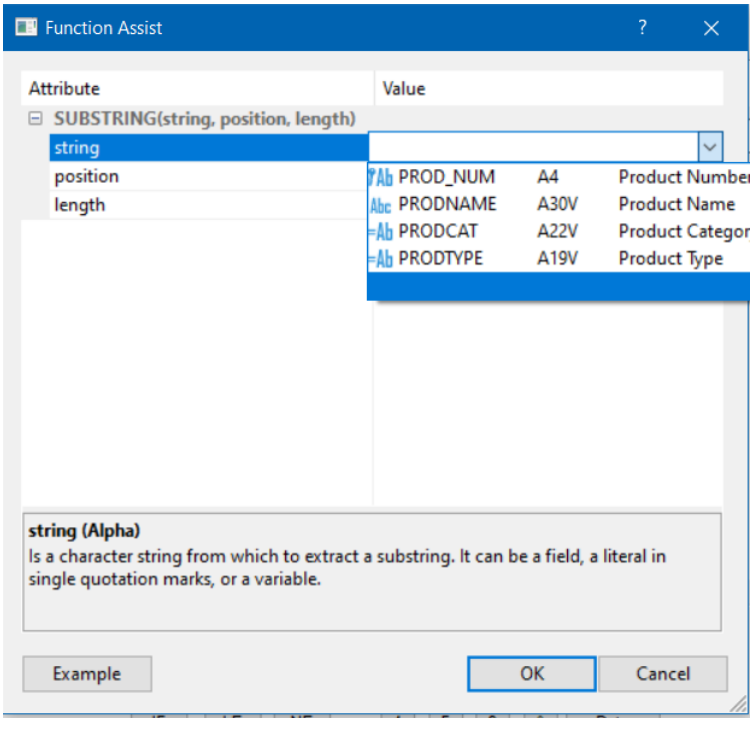
Procedure: How to Use Function Assist

When you use a function in one of the calculators, the Assist dialog box can help you set the parameters.

1. Click the *Functions* tab in one of the calculators.
2. Double-click a function to add it to the Expression tab.

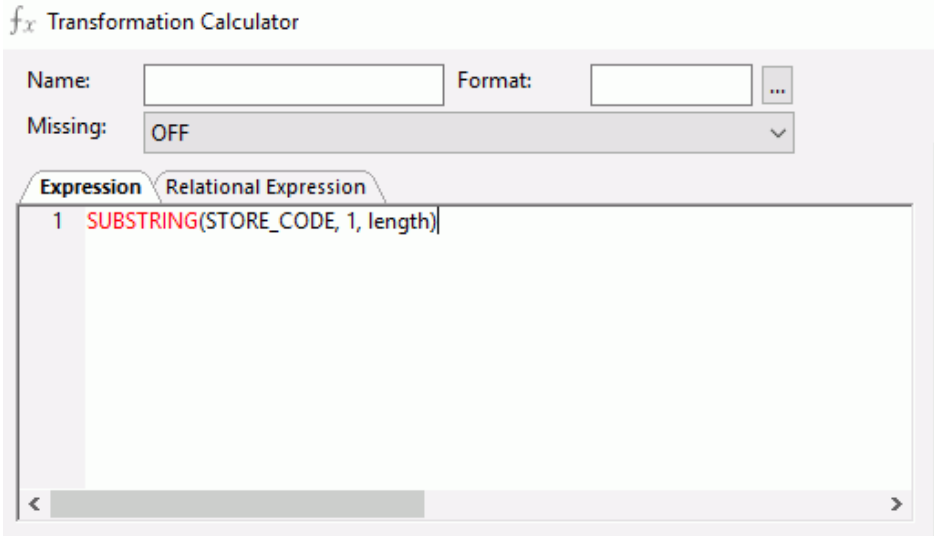
   Note: Right-click the *Name* heading bar to display the functions by Tree View or in List View.

   The Function Assist dialog box opens.
3. Enter alphanumeric parameters in their fields.
4. Enter parameters that require column names by selecting them from the drop-down menu, as seen in the following image.



   In addition to the column name, the format, description, and nullability are also displayed.
5. When all the parameters in the Function Assist dialog box have been set, click *OK*.

The function is added to the Expression tab of the calculator, as shown in the following image.



Note: The Function Assist button enables you to edit a function in the Expression tab using the Function Assist dialog box.

Column Formats and Mapping

When you add transformations using either the Source or Target Transformations window, you can assign formats to the columns that you are adding. There are four types of column formats:

- **Alphanumeric formats.** Composed of letters, numbers, or special characters. There are three alphanumeric formats:
  - **Fixed Alphanumeric.** Used for fixed-length columns. The maximum length is 256.
  - **Variable Alphanumeric.** Used for variable-length columns. The maximum length is 4,096.
  - **Text.** Used for text columns, such as MS SQL Server VARCHAR(MAX) or an ORACLE CLOB.
  - **String.** Used for columns in Relational databases that have a STRING format, which supports text data of unlimited length.

- **Numeric formats.** Composed of the digits 0 through 9 and, optionally, a minus sign. There are four numeric formats:
  - **Integer.** Used for whole numbers, using 4 bytes of storage and a maximum number of 9 positions. Numbers after the decimal are truncated.
  - **Float.** Used for whole numbers or fractions, using 4 bytes of storage and a maximum number of 9 positions.
  - **Double.** Used for whole numbers or fractions, using 8 bytes of storage and a maximum number of 15 positions.
  - **Decimal.** Used for whole numbers, using one byte for each two digits, plus a sign and a maximum number of 15 positions. This type of column will automatically round any number with more than the specified number of decimal places to the number of decimal places in the format. This is recommended for monetary values.

  For more information about numeric formats, see *Numeric Format Options* on page 348.
- **Date and time formats.** Composed of dates and times or date and time components. There are three formats:
  - **Date.** Used for dates and provides combinations of year, month, day, and quarter.
  - **Time.** Used for times and provides combinations of hour, minute, second, and millisecond.
  - **DateTime (Timestamp).** Used for both dates and times and provides combinations of year, month, and day for dates, and hour, minute, second, millisecond, and microsecond for time.
- **Hexadecimal.** Displays the hexadecimal representation of a field value.
- **BLOB.** Represents a Binary Large Object. A BLOB is used to store an image, sound, or some other large, unstructured block of data. The corresponding database data types are:
  - ORACLE - BLOB
  - MS SQL Server - Varbinary (max)
  - Informix - Byte
  - Db2 - BLOB

  A column described as BLOB can only be copied from source to target. No transformations can be performed.

Reference: Rules for Mapping and Column Formats

Columns that you map to each other must have the same format, or Data Migrator must be able to translate the source query format into the target format automatically. Details are shown in the following table:

If the Target format is...	...and the Source query format is...	Can you map?
A11	A10	Yes
A10	A11	No. Use the EDIT or SUBSTR functions to make this work.
D6.2	I5	Yes
I10	D6.2	Yes, but you will lose the numbers after the decimal.
A10	D6.2	No

Procedure: How to Format a Column as Alphanumeric or Numeric

If you are creating a new or virtual column, you need to assign a format to the column. You can assign a format in two ways:

- By typing the format into the Format field in the Transformation Calculator.
- By using the Define Properties dialog box.

To use the Define Properties dialog box:

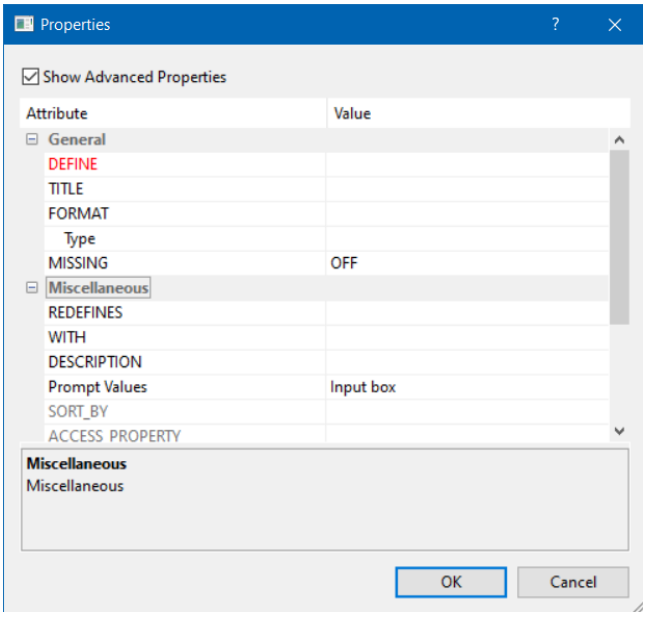
1. From the Transformation Calculator, click the ellipsis (...) button to the right of the Format field.

   The Define Properties dialog box opens.

2. Select a format from the Type drop-down menu.
3. Click *OK*.

Reference: Define Properties Dialog Box

To access the Define Properties dialog box from the Transformation Calculator, click the ellipsis (...) button to the right of the Format field. To see the Miscellaneous properties, select the *Show Advanced Properties* check box.



The Define Properties dialog box has the following fields and options:

DEFINE

An optional attribute that is used to create a virtual field for reporting.

FORMAT

The format name of the column.

Type

The format type of the column.

Note: The options available will depend on the type of format selected.

MISSING

Indicates if the results can be missing. The following options are available:

- **OFF.** The result is never missing.
- **ON.** The result is missing only if all missable fields are missing.

- **ON ALL.** The result is missing if any missable fields are missing.

REDEFINES

Lets you redefine or recompute a field whose name exists in more than one segment.

WITH

Lets you identify the logical home of the defined calculation, if no fields from the synonym are used in the expression or have been defined.

DESCRIPTION

An optional attribute that enables you to provide comments and other documentation for a field within the Master File. Comments can be up to 2048 characters in length.

Prompt Values

Optional accept values for the define. Choices are:

- **Input box.** The user enters a value.
- **Static list of values.** A field appears with an ellipsis. When you click it, a dialog box opens for adding values.
- **Synonym based dynamic list of values.** Fields open for entering a Lookup Synonym, Lookup Field, and Lookup Display Field.
- **Procedure based dynamic list of values.** Fields open for entering a Lookup Procedure, Lookup Procedure Parameters, Lookup Field, and Lookup Display Field.

SORT_BY

Is the name of the sort by field.

ACCESS_PROPERTY

To make this a field that will not display in sample data or in the list of available fields, select the *INTERNAL* check box.

GEOGRAPHIC_ROLE

Assigns a geographic role to the field. Choices are:

- **ADDRESS_FULL.** Full address.
- **ADDRESS_LINE.** One line of address.
- **CITY.** City name.
- **CONTINENT.** Continent name.
- **COUNTRY.** Country name.
- **COUNTY.** County name.

- **GEOMETRY_AREA.** JSON geometry area.
- **GEOMETRY_LINE.** JSON geometry line.
- **GEOMETRY_POINT.** JSON geometry point.
- **LATITUDE.** Latitude.
- **LONGITUDE.** Longitude.
- **POSTAL-CODE.** Postal code.
- **STATE.** State name.

TEMPORAL_PROPERTY
If the field has a temporal property, select *Trend*.

USE_STYLE
Defines font, font style, and color properties for the field.

Reference: Numeric Format Options

When using a numeric format (integer, float, double, or decimal packed), the following options are available:

Negative

These are the display options for negative numbers:

Default - Minus sign at the left

Positions the minus (-) sign at the left of the number.

- - Minus sign at the right

Positions the minus (-) sign at the right of the number.

B - Bracket negative

Encloses a negative number in parentheses.

C - Credit (CR) negative

Adds the characters CR after a negative number.

Comma

These are the display options for commas:

Default - Suppress

Suppresses the use of a comma in the number.

c - Suppress

Suppresses the use of a comma in the number.

C - Insert

Inserts a comma after every third significant figure. This option takes effect automatically for Double format.

L - Leading zeroes

Adds leading zeros to the full field length.

S - Print blank for zero

Suppresses leading zeros; displays a space if the value is 0.

Abbreviated Display

Choices are:

- **k.** Displays the number in thousands, for example. 12.25K.
- **m.** Displays the number in millions, for example. 12.25M.
- **b.** Displays the number in billions, for example. 12.25B.
- **t.** Displays the number in trillions, for example. 12.25T.
- **a.** Displays the number automatically using the most useful abbreviation.

Percent Sign

Select either %, which displays a percent sign along with the numeric data, or *p*, which calculates the percent and adds a percent sign.

E - Scientific notation

Displays the number in scientific notation.

Currency Symbol

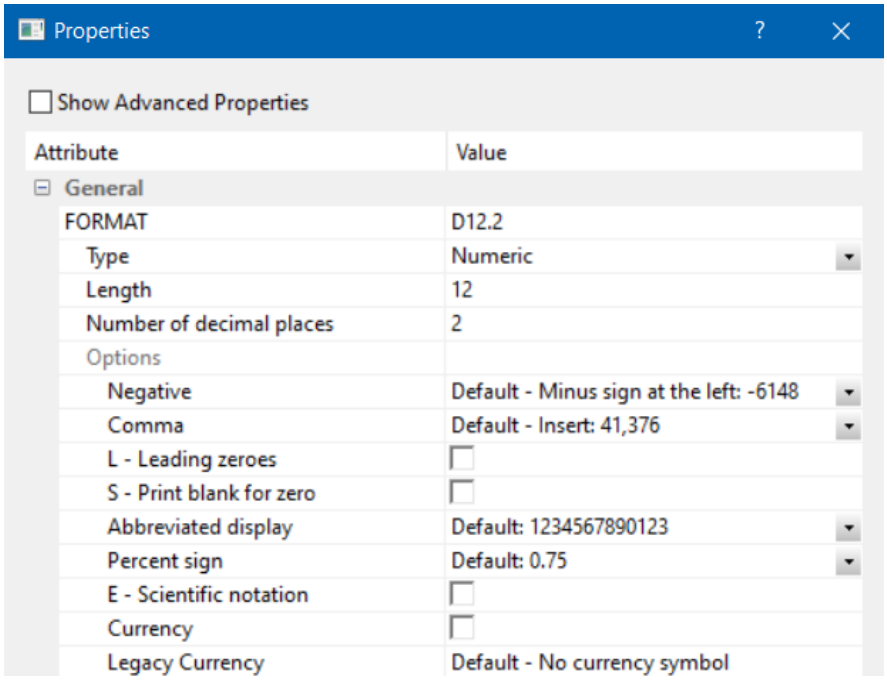
These are the extended currency symbol display options, which allow you to select a currency symbol regardless of what is set as the default:

- N - Fixed $ sign
- M - Floating $ sign
- !e - Fixed euro symbol
- !E - Floating euro symbol
- !l - Fixed pound sterling sign
- !L - Floating pound sterling sign
- !y - Fixed Japanese yen symbol
- !Y - Floating Japanese yen symbol
- !d - Fixed dollar sign
- !D - Fixed dollar sign

Currency

Specifies a field as a currency denominated value.

The Define Properties dialog box for the Numeric format is shown in the image below:



Procedure: How to Format a Column as Date, Time, or DateTime (Timestamp)

1. In the Transformation Calculator, click the ellipsis (...) button to the right of the Format field.

   The Define Properties dialog box opens.

2. Select Date, Time, or Date and Time from the FORMAT drop-down menu.

   The Date format allows you to choose the display for dates (YYMD, MDY, and so on).

   The Time format allows you to choose the display for time (seconds, milliseconds, and so on) and am/pm.

   The Date and Time format allows you to choose the display for both date and time portions.

Reference: Date Type Format Options

The following choices are available from the Define Properties dialog box for the date format (the examples illustrate August 21, 2004):

Format	Description	Example
YYMD	Four-digit year/month/day.	2004/08/21
MDYY	Month/day/four-digit year.	08/21/2004
DMYY	Day/month/four-digit year.	21/08/2004
YYQ	Four-digit year/quarter	2004 Q4
QYY	Quarter/four-digit year.	Q4 2004
YYM	Four-digit year/month.	2004/08
MYY	Month/four-digit year.	08/2004
YYJUL	Four-digit year with the month in Julian format. The last three digits are the number of days since January 1.	2004/234
YY	Four-digit year.	2004
Y	Two-digit year.	04
Q	Quarter.	Q3
M	Month.	08
MT	Short month in uppercase.	AUG
Mt	Short month in mixed-case.	Aug
MTR	Full month in uppercase.	AUGUST
Mtr	Full month in mixed-case.	August
W	Three-letter day of the week abbreviation in uppercase.	SAT
w	Three-letter day of the week abbreviation in mixed-case.	Sat
WR	Day of the week in uppercase.	SATURDAY
wr	Day of the week in mixed-case.	Saturday

Format	Description	Example
D	Day.	21

Reference: **Time Type Formats**

The following choices are available from the Define Properties dialog box for the time format (the example illustrates 2:05 A.M.):

Format	Description	Example
HH	Hours.	02
HHI	Hours/Minutes.	02:05
HHIS	Hours/Minutes/Seconds.	02:05:27
HHISs	Hours/Minutes/Seconds/Milliseconds.	02:05:27:123
HHISsm	Hours/Minutes/Seconds/Milliseconds/Microseconds.	02:05:27:123456
HI	Minutes.	05
HIS	Minutes/Seconds.	05:27
HISs	Minutes/Seconds/Milliseconds.	05:27:123
HISsm	Minutes/Seconds/Milliseconds/Microseconds.	05:27:123456
HS	Seconds	27
HSs	Seconds/Milliseconds.	27:123
HSsm	Seconds/Milliseconds/Microseconds.	27:123456
Hh	Hours without leading zeros.	2
HhI	Hours without leading zeros/Minutes.	2:05
HhIS	Hours without leading zeros/Minutes/Seconds.	2:05:27
HhISs	Hours without leading zeros/Minutes/Seconds/ Milliseconds.	2:05:27:123

Format	Description	Example
HhISsm	Hours without leading zeros/Minutes/Seconds/ Milliseconds/Microseconds.	2:05:27:123456
Hi	Minutes without leading zeros.	5
HiS	Minutes without leading zeros/Seconds.	5:27
HiSs	Minutes without leading zeros/Seconds/Milliseconds.	5:27:123
HiSsm	Minutes without leading zeros/Seconds/Milliseconds/ Microseconds.	5:27:123456

Reference: **Date and Time Type Formats**

The Date and Time type format combines date, time, and a separator, which are reflected in the format name. For example, HYYMD-H describes a four-digit year, month, and day separated by dashes plus hours.

The following choices are available from the Define Properties dialog box for the date component:

Format	Description	Example
HYYMD	Four-digit Year/Month/Day.	2004/08/21
HYMD	Two-digit Year/Month.	04/08
HMD	Month/Day.	08/21
HD	Day	21

The time component is added to the date format name as follows:

Format	Description	Format Name	Example
H	Hours.	HYYMDH	2004/08/21 02
I	Hours/Minutes.	HYYMDI	2004/08/21 02:05

Format	Description	Format Name	Example
S	Hours/Minutes/Seconds.	HYYMDS	2004/08/21 02:05:27
s	Hours/Minutes/Seconds/Milliseconds.	HYYMDs	2004/08/21 02:05:27:123
m	Hours/Minutes/Seconds/ Milliseconds/Microseconds.	HYYMDm	2004/08/21 02:05:27:123456

The following choices are available from the Define Properties dialog box for the separator, which is added to the format name.

Format	Description	Format Name	Example
/	Slash.	HYYMDH	2004/08/21 02
.	Period.	HYYMD.H	2004.08.21 02
-	Dash.	HYYMD-H	2004-08-21 02
Blank	Space	HYYMDBH	2004 08 21 02
None	No separator	HYYMDNH	20040821 02
U	International default.	HYYMDU	

Setting Properties of a Flow

The properties of a flow allow you to view or set properties that apply to the entire flow. These include:

- General properties, including who created the flow, the last time it was updated, a description, and comments.
- Execution properties, including stop processing options and the scheduler status. (Scheduler status is used for process flows.)
- Restart properties, including the number of restart attempts and their start points. A restart attempt occurs if the flow execution fails. Restart attempts can be made from the first record or from the last commit point.

- Email notification properties, including setting different criteria for sending email to a specific address.
- Comments, which can be helpful as reminders, or to pass information among a group of users working on the same flows.

For more information on how to set logging properties for a flow, see *Setting Flow Properties* on page 144.

Running a Data Flow

There are two main ways to run a data flow from the DMC: Run and Submit.

Run will run the data flow immediately. When you run a flow, processing information is displayed in the Console Log.

Submit will submit the data flow to the scheduler to run. When you submit a flow, the processing information is recorded in the Data Migrator Log and Statistics tables. Email messages are also sent if the flow properties are set to send email messages at the start or end of the job.

Procedure: How to Run a Data Flow

On the Flow tab, in the Run group, click *Run*, and select one of the run options available.

or

Right-click a data flow in the navigation pane and select one of the Run options available.

Note: When you submit the flow, processing information is recorded in the DM log and, if specified, email messages are sent. When you run the flow, the processing information is only displayed in the Console Log. It is *not* written to the DM log.

For more information on the Run options, see *Procedures Shortcut Menu Options* on page 114.

Using the Cluster Manager Scheduler

The Data Migrator scheduler can be used to automatically initiate Data Flows at a specified time or interval. However, if you are connected to a single server and it is not running, those flows will not execute when they are scheduled to run.

The Cluster Manager (CLM) available in the DMC results in a highly available Data Migrator scheduler that automatically switches to other application servers in the event that the main server is unavailable.

CLM regulates the server nodes of the configured cluster and makes a decision about which cluster node should run the scheduled jobs. One of the advantages of CLM is that it can help with business continuity tasks. You can create a Cluster and assign multiple nodes on the CLM. With CLM configured, the scheduled jobs run on the nodes that were added to the Cluster. The jobs scheduled on each node will be running on one node only; the currently active node, which is the node that has the Scheduler running.

To use the Cluster Manager, the following is required:

- ❑ A server to use as the Cluster Manager.
- ❑ Two or more Reporting Servers.
- ❑ An Application path for ScanID that is copied or shared.

Procedure: How to Enable the Cluster Manager

To enable the Cluster Manager from the DMC, connect to your server and perform the following steps.

1. Expand the *Workspace* folder.
2. Right-click *Cluster Manager* and click *Enable*.
3. Select Cluster Manager properties, or accept the defaults, and click *Save and Restart Server*.

   You have the option of using dynamic load balancing in a multi-cluster environment, by selecting *Yes* in the drop-down box. This configuration is designed for scalable cloud environments, but can also be used for on-premise environments. It requires external load balancing software (such as NGNIX) and ETCD (for inter-communication of server state and status) to be installed, running, and accessible. If you are using Multi-CLM with Dynamic Load Balancing mode, the first step is to install and start your load balancing software and ETCD. They do not need to be on the machine where the server is running, nor do the balancer and ETCD need to be on the same machine, but you will need to know the machine node names and the ports on which they (the balancer and ETCD) are listening, and have a read/write access ID and password for ETCD. You need to enter values for the following properties.

   ETCD URL
   Enter node and port in the format http://node:port, or https://node:port, depending on whether your ETCD configuration is configured for SSL.

   ETCD Name
   Enter an ETCD user ID that has Read/Write access.

ETCD Password

Enter the password associated with the user ID entered in the ETCD Name entry field.

ETCD Key

Enter an arbitrary string (for example, Accounting). This value acts as a key, and allows the ETCD Server to be reused for multiple groups of Multi-CLM Severs, while eliminating cross-cluster noise between the unrelated CLM Servers. If you are only using a single group of CLM Servers in your ETCD environment, you can leave this field blank. However, supplying a value provides added identification when examining ETCD records, so it is highly recommended to supply a value.

4. Click *OK* to submit your changes and restart the server.

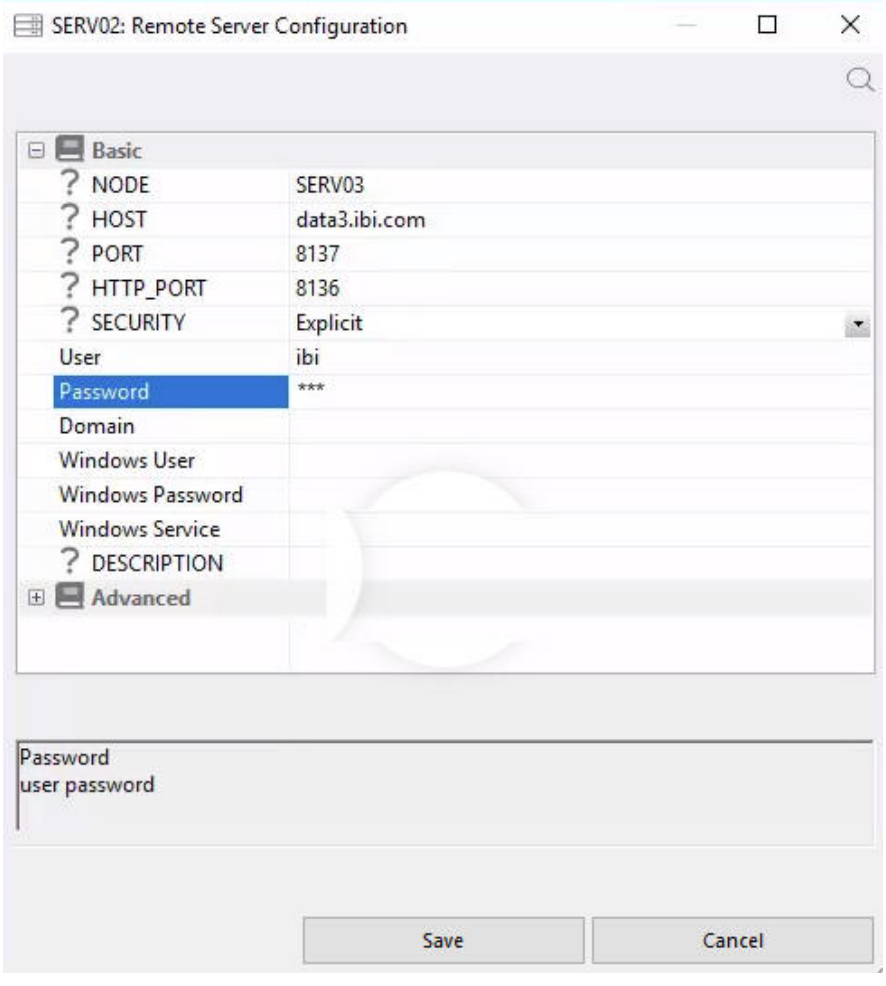
Procedure: How to Add Remote Nodes

To add nodes for cluster management:

1. From the Workspace folder, right-click *Cluster Manager*, point to *New Remote Server*, and click *TCP*.

   The Remote Server Configuration dialog box opens.

2. Enter the configuration settings for the first Reporting Server, as shown in the following image.



3. Click *Save*.

Procedure: How to Configure a Cluster

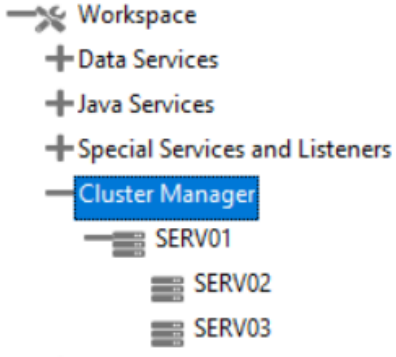
To configure a cluster that includes your remote nodes:

1. From the Workspace folder, right-click *Cluster Manager*, point to *New Remote Server*, and click *Cluster*.

   The Remote Server Configuration dialog box opens.

2. Enter the configuration settings for the cluster and click *Save*.

   The nodes that comprise the cluster will appear under the cluster name, as shown in the following image.



To view or edit the properties for the cluster manager, right-click *Cluster Manager* and click *Properties*. You can edit the properties as needed.

Notes:

- All servers on the assigned nodes should run the same version of Data Migrator.
- Scanning on all nodes should be enabled. To enable the Scheduler to scan the parameters, set *sched-autostart* to *Y* in the Scheduler Properties dialog box.
- The scheduler should be stopped on all nodes except for one active node. When the Cluster Manager finds that the Scheduler is running on another node other than that active node, it attempts to stop the Scheduler on that node. If a server on an active node goes down, the Cluster Manager detects this and will start the Scheduler on the next available server node.
- Only a Server Administrator or Server Operator can start or stop the Schedule.

Procedure: How to View the Activities on a Cluster

Once jobs are scheduled on each server node, you can review the statistics of your configured cluster.

1. Right-click the cluster name and click *Statistics*.

   The CLM Node Statistics dialog box opens showing the details of the cluster. The Active Scheduler parameter reflects the name of the node that is currently active

2. To refresh the connection, re-connect to your server node where your Cluster Manager is running.

3. Expand the *Workspace* folder, followed by the *Logs and Traces* folder to view the Cluster Manager Log.

Migrating Data Flows

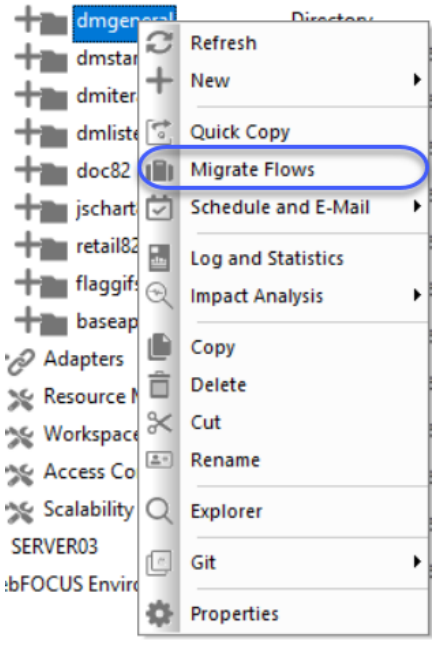
When you create a Data Flow, an SQL request is generated that retrieves data from Source databases, applies transformations, if necessary, and loads the data to Target databases.

In new releases of Data Migrator, improvements are made, when possible, to the code that is generated for the Flows. The new code uses new back-end algorithms for reading and loading the data, allowing an improved performance.

If a Data Flow created in a prior release no longer runs, or seems to run too slowly, you can migrate the flow to the current release. This process will update the code for the generated Flow to the latest syntax, and take advantage of performance and optimization improvements available in the current release of Data Migrator.

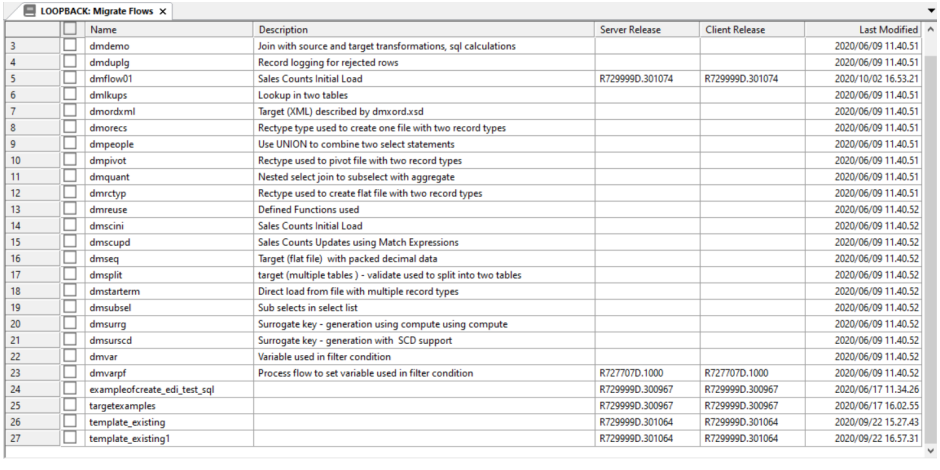
Procedure: How to Migrate Data Flows

1. Right-click an application directory, and click *Migrate Flows*, as shown in the following image.



Note: If you do not see the see the menu item *Migrate Flows* on the Application Directory context menu, you can enable it from the Preferences pane. Right-click the server node name and select *Preferences* from the context menu to open the Preferences pane. Expand the Technical Preview Features section. Select *Yes* from the *Enable Migrate Flows* drop-down list.

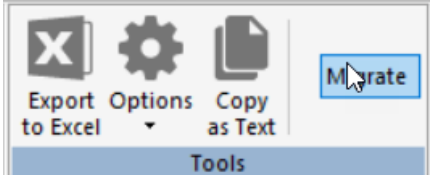
The Migrate Flows panel opens in a new tab, as shown in the following image.



LOOPBACK: Migrate Flows

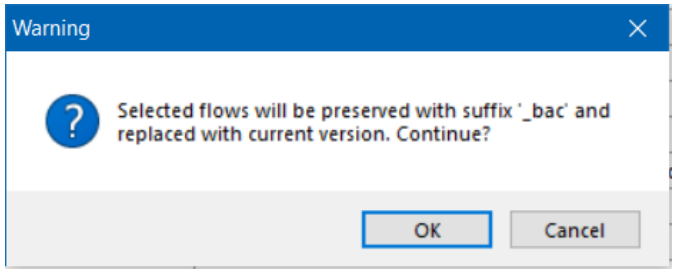
		Name	Description	Server Release	Client Release	Last Modified
3	☐	dmdemo	Join with source and target transformations, sql calculations			2020/06/09 11.40.51
4	☐	dmduplg	Record logging for rejected rows			2020/06/09 11.40.51
5	☐	dmflow01	Sales Counts Initial Load	R729999D.301074	R729999D.301074	2020/10/02 16.53.21
6	☐	dmlkups	Lookup in two tables			2020/06/09 11.40.51
7	☐	dmordxml	Target (XML) described by dmxord.xsd			2020/06/09 11.40.51
8	☐	dmorecs	Rectype type used to create one file with two record types			2020/06/09 11.40.51
9	☐	dmpeople	Use UNION to combine two select statements			2020/06/09 11.40.51
10	☐	dmpivot	Rectype used to pivot file with two record types			2020/06/09 11.40.51
11	☐	dmquant	Nested select join to subselect with aggregate			2020/06/09 11.40.51
12	☐	dmrctyp	Rectype used to create flat file with two record types			2020/06/09 11.40.51
13	☐	dmreuse	Defined Functions used			2020/06/09 11.40.52
14	☐	dmscini	Sales Counts Initial Load			2020/06/09 11.40.52
15	☐	dmscupd	Sales Counts Updates using Match Expressions			2020/06/09 11.40.52
16	☐	dmseq	Target (flat file) with packed decimal data			2020/06/09 11.40.52
17	☐	dmsplit	target (multiple tables) - validate used to split into two tables			2020/06/09 11.40.52
18	☐	dmstarterm	Direct load from file with multiple record types			2020/06/09 11.40.52
19	☐	dmsubsel	Sub selects in select list			2020/06/09 11.40.52
20	☐	dmsurrg	Surrogate key - generation using compute using compute			2020/06/09 11.40.52
21	☐	dmsurscd	Surrogate key - generation with SCD support			2020/06/09 11.40.52
22	☐	dmvar	Variable used in filter condition			2020/06/09 11.40.52
23	☐	dmvarpf	Process flow to set variable used in filter condition	R727707D.1000	R727707D.1000	2020/06/09 11.40.52
24	☐	exampleofcreate_edi_test_sql		R729999D.300967	R729999D.300967	2020/06/17 11.34.26
25	☐	targetexamples		R729999D.300967	R729999D.300967	2020/06/17 16.02.55
26	☐	template_existing		R729999D.301064	R729999D.301064	2020/09/22 15.27.43
27	☐	template_existing1		R729999D.301064	R729999D.301064	2020/09/22 16.57.31

2. Select the check boxes next to the flows you want to migrate, and click *Migrate* in the Tools group on the ribbon, as shown in the following image.



Optimize Load will be used whenever it is supported for the Target DBMS.

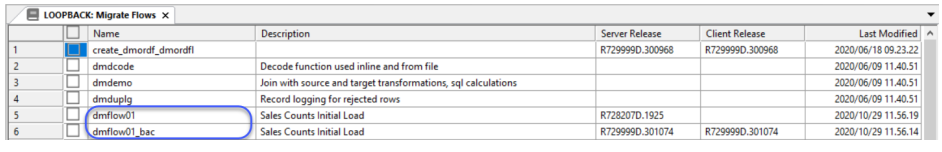
The original versions of the selected flows will be saved with the suffix _bac appended to the flow name. The migrated flows will have the original names. The following message displays to inform you of this.



If you receive this message, either click *OK* to migrate the flows and save the original versions with names that have the suffix _bac, or click *Cancel* to cancel the migration.

3. Close the Migrate Flows tab by clicking the X.

   The original flows (with the _bac suffix) and migrated flows (with the original names) will appear on the panel, as shown in the following image.



Creating a Direct Load Flow

A Direct Load flow is a simplified flow that can be used for a single source. It loads data directly from the file into an existing target, without the need to create an intermediate file. This occurs unless Bulk Load from Disk File is used, which always creates an intermediate file.

Note: The synonym that describes the source table for a Direct Load Flow must contain a unique ALIAS that matches the FIELD name for every column that is referenced.

A Direct Load flow can load from hierarchical data sources, such as VSAM. For example, a file with multiple record types can be used as a source with fields from each record type loaded into a separate table.

A Direct Load flow has no SQL object, so no aggregation (group by), filtering (where conditions), or SQL calculations can be performed.

Source transformations and virtual columns in the source synonym cannot be used.

Target transformations, functions, variables, validates, target properties, and record logging can be used. The procedures for using these capabilities, as well as for running the flow, are the same for Direct Load flows as for more typical data flows. For information, see the appropriate topics earlier in this chapter.

While other types of data flows discard records with format errors, when a direct load flow runs, any records rejected due to format errors are counted, and the number of rejected rows appears in the log. Optionally, the rejected records can be written to a log file.

For information, see *How to Set the Data Flow Record Logging Options* on page 288.

Note: Support for using Direct Load Flows with a source and target tables in the same relational database, and using insert records from memory, depends on the characteristics of the database. For MS SQL Server JDBC, ORACLE, and Teradata, there are no restrictions. For MS SQL Server OLE DB, Sybase, and Informix, use a Commit number greater than the number of rows to be loaded. For Db2, use separate connections for source and target.

Procedure: How to Create a Direct Load Flow

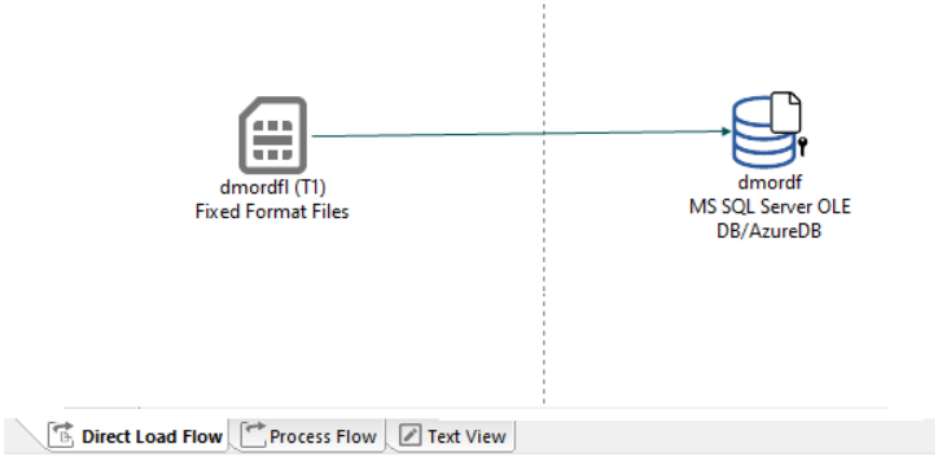
1. Right-click the application directory where you want the new data flow, point to *New*, and then click *Direct Load Flow*.

   An empty flow appears in the data flow workspace. The Direct Load Flow tab opens by default.

   Note: An SQL object does not appear in the center of the workspace.

2. Drag a data source into the workspace to the left of the center line.
3. Drag a data target into the workspace to the right of the center line. You can also right-click in the workspace to the right of the center line, point to *Add Target*, and then click *Existing*.
4. Set up target transformations. For information, see *Mapping Transformations* on page 305.
5. Right-click a target object and click *Properties*. Select the load type from the drop-down menu.
6. Click the *Save* button on the Quick Access Toolbar.
7. Enter a name for the flow in the *Name* field and click *OK*. You can also select a Server and Application Directory from the drop-down menus.
8. Run the flow.

An example of a Direct Load flow is shown in the following image.



Creating a DBMS SQL Flow

A DBMS SQL flow allows you to enter dialect-specific SQL SELECT statements for extracting data from a relational database. Dialect-specific SQL is the SQL specific to one particular relational database, such as ORACLE or MS SQL Server.

You can type in the SELECT statement, or import it from a pre-existing .sql file.

A DBMS SQL flow only contains a specialized SQL object where you select the DBMS adapter and connection, and one or more target objects. Synonyms for sources are neither required nor used. Tables must be identified using the same names as in the database.

While source transformations cannot be used, target transformations, functions, variables, validates, target properties, and record logging can be used. The procedures for using these capabilities are the same for DBMS SQL flows as for more typical data flows.

For information on these procedures, see the appropriate topics earlier in this chapter.

Procedure: How to Create a DBMS SQL Flow

The following procedure uses the Microsoft SQL Server pubs.dbo.authors sample table, and the Microsoft SQL Server difference scalar function, which is not available in ANSI SQL.

1. Right-click the application directory where you want the new data flow, point to *New*, and then click *DBMS SQL Flow*.

   An empty flow appears in the data flow workspace. The DBMS SQL Flow tab opens by default.

A DBMS SQL Object appears in the center of the workspace.

2. Right-click the DBMS SQL object and click *Properties*.
3. Select an adapter and connection from the drop-down menus and close the Properties panel.
4. Right-click the DBMS SQL object and click *SQL Statement*.

   The Select Statement dialog box opens.
5. Type a dialect-specific SQL Statement, as shown in the following Select Statement dialog box.

```
SELECT
  au_fname, au_lname
FROM
  pubs.dbo.authors
WHERE
  difference  (au_lname, 'delay')  > 3
```

6. Test your SQL by clicking the *Test SQL Statement* button. The results are shown in the following image.

	au_fname	au_lname
1	Innes	del Castillo
2	Ann	Dull

7. Click *Close*. Then click *OK*.

   Note: The SQL statement must be a SELECT statement that conforms to the dialect-specific requirements of the adapter.
8. Optionally, you can import an SQL statement from an existing .sql file by clicking the *Import* button, pointing to an SQL file, and clicking the *Select* button. The SQL file must be in one of your Application Directories.
9. Optionally, you can export SQL statements as .sql files for later use by clicking the *Export* button. Select a Server and Application Directory from the drop-down menus, enter a File Name, and click *OK*.
10. Drag a data target into the workspace. You can also right-click in the workspace, point to *Add Target*, then *Existing* or *Add Target*, and then click *New*.
11. Click the *Save* button on the Quick Access Toolbar.
12. Enter a name for the flow in the *File Name* field and click *OK*. You can also select a Server and Application Directory from the drop-down menus.
13. Run the flow.

Creating a Quick Copy

The *Quick Copy* option generates a simple Data Flow without using the Data Flow designer. For the selected tables, all selected rows and columns are copied to new tables.

Although no transformations or where conditions are generated, the Data Flow can be opened as such from the Data Management Console, and additional criteria can be added. Columns identified as key columns in the source are created as key columns in the target.

Procedure: How to Create a Quick Copy for a Single Table

1. Right-click a synonym name and click *Quick Copy*.

   The Quick Copy dialog box opens. The following options are available:

   Load Option

   - **New.** Any existing metadata or data is replaced.
   - **Append Existing Data.** The source data is appended to the target.

   Adapter

   Use the drop-down menu to select a database configured as an adapter.

   Connection

   Use the drop-down menu to select a connection for the selected adapter.

   Synonym Application

   The application directory where the target tables and generated flow will be stored.

   Synonym

   The synonym name for the target table. By default, this is the same name as the source table unless the Application directory is the same, in which case _t is added to the name.

   Table Name

   The name of the target table name in the database. By default, this is the same name as the source table, unless the Adapter and Connection are the same as the source, in which case _t is added to the name.

   Bulk Load

   Specifies the method of loading data. When selected, the database vendors bulk load program is used, instead of inserting records from memory.

2. In the Target Synonym Application box, select the application directory where to save the flow by either typing in the location of the directory, or clicking the ellipses and manually selecting it from the Select Application dialog box.

3. Click *Save* or *Run*. Two flows will be created, a process flow with the name you specified, and a data flow with the name followed by the table name, for example, flow01_dmhr. The process flow calls the data flow to copy the table.

Procedure: How to Create a Quick Copy for Multiple Tables

1. Right-click an application directory and click *Quick Copy*.

   The Quick Copy dialog box opens.

2. Select the tables you want to copy by selecting the check box in the first column. Doing so allows you to change the Name, which is the synonym, and the Table Name, which is the name of the table in the database.

   Note: If you select the Parallel Execution checkbox, the flows for each table will be run in parallel.

3. Click *Save* or *Run*. Two or more flows will be created, a process flow with the name you specified, and a data flow with the name followed by the table names for each synonym you have selected. For example, flow01_dmreps. The process flow calls all of the data flows.

Chapter 6

Designing a Process Flow

A process flow defines a set of steps that execute a series of components or objects, including the data flows that define your extractions, transformation, and load logic. You can combine:

- A Start object with a schedule that defines when to run the process.
- Data Flow objects that indicate what data to extract and move from data sources to data targets.
- Email objects that notify users about the status of the process at specified points.
- Procedure objects that perform a variety of supplementary tasks.
- Connector objects, represented as arrows, that specify execution logic for the other objects included in the process flow.
- Variable objects to set the values for variables used in other objects.
- Group objects that specify the flow of a subset of objects within the total flow.

In this chapter:

- Process Flow Methodology
- Process Flow Interface
- Simple Process Flow At a Glance
- Complex Process Flow at a Glance
- Creating a Process Flow
- Adding a Saved Procedure to a Process Flow
- Assigning and Using Variables in a Process Flow
- Maintaining Continuous Processing in a Single Agent
- Executing a Process Flow That Contains Only Procedures
- Adding Email Messages to a Process Flow
- Setting Connection Logic
- Grouping and Controlling Simultaneous Execution of Flows
- Running a Process Flow
- Scheduling a Process Flow

Process Flow Methodology

You can create a process flow either with or without a data flow. You can open a data flow from within a process flow by right-clicking it and selecting *Open*.

Creating a Process Flow With a Data Flow

Although you can build a simple process flow as an extension to an existing data flow, it is recommended that you create and save each flow and procedure separately. With this modular approach, individual flows (both data and process flows) and procedures can be executed from more than one process flow, and a process flow can be modified as needed and used to incorporate different data flow and procedure objects. A change to a data flow or procedure automatically affects all of the process flows that use it.

If you click the Text View tab in a process flow created with this method, any data flows or procedures used in the process flow appear as EX procedure_name. If you want to view or edit the underlying code for an individual procedure, you can right-click the object and click *Edit as Text*.

However, if you want to add a few process flow objects, such as a schedule or an email, to a data flow, and do not require the advantages of a modular design, save the data flow and click the Process Flow tab, and then add the process flow components to the existing data flow.

Note: If you click the Text View tab in a process flow created with this method, the underlying data flow and process flow syntax appears fully, since it is part of the same procedure.

Creating a Process Flow Without a Data Flow

Normally, process flows manage extraction, transformation, and load processing. However, process flows are not limited to this functionality. Under some circumstances, you may wish to create process flows that do not contain any data flows.

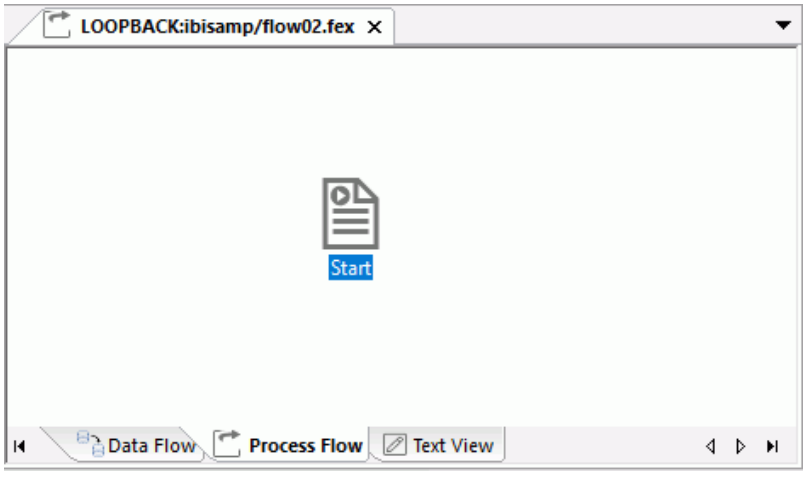
Using the same set of design tools, you can, for example, create a process flow that launches a series of reporting procedures on a defined schedule, and then sends email messages to appropriate recipients when the procedures are complete.

Process Flow Interface

The Process Flow interface consists of:

- The Process Flow tab and workspace, where you build your process flows.
- The active Flow tab, which enables you to add objects to your process flow.

When you create a new flow and click the Process Flow tab in the workspace, a Start object appears in the workspace to anchor the flow, as shown in the following image.



You can add a variety of other objects from the Insert group. These may include data flows or other process flows, as single components or as a group, procedures, and email messages, in whatever combination and order suits your needs.

The remainder of this chapter describes, in detail, all the options that are available to you as you design your process flows.

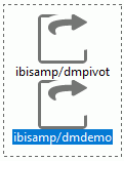
For more information on using the Procedures button, see *Procedures Shortcut Menu Options* on page 114.

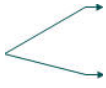
Reference:

Process Flow Objects in the Workspace

Process flow objects appear as follows in the workspace:

Object	Function
	Start Indicates the start of the process flow. Start is always the first object in the process flow. The Start object contains the scheduler attributes for the process flow. You can schedule a flow to run once, on a recurring basis, or on a specified days of the week or month, on an interval you define.

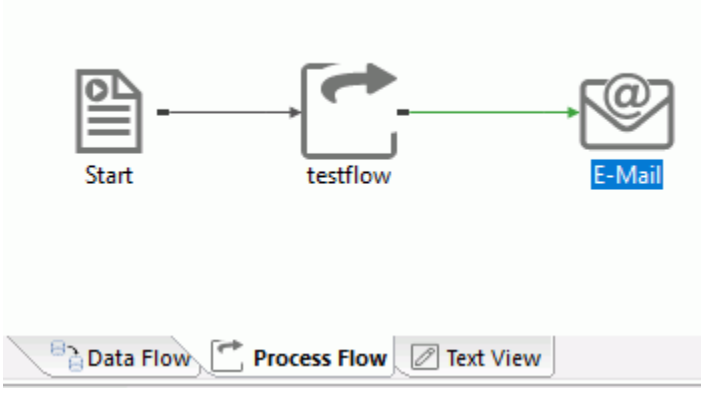
Object	Function
	Data Flow Indicates an embedded data flow that contains data source and data target objects, along with business rules that govern what is extracted and moved from source to target objects. To open the data flow, right-click the flow and select *Open*. **Note:** This icon appears when the data flow is embedded in a process flow.
	Flow Indicates that a complete data flow or another process flow is being executed from the current process flow. You can nest one flow within another. To open the flow, right-click the flow and select *Open*. **Note:** This icon appears when a previously saved data flow or process flow is executed from the current process flow.
	Parallel Group Indicates that a group of flows or procedures will be executed simultaneously. The group object is generally followed by a Wait object.
	Wait Indicates that processing of the next object in the flow will wait until execution of all processes in the parallel group have been completed. A Wait object generally follows a Parallel Group object, but is not required.

Object	Function
	E-mail Indicates an email message is sent at the designated points in the flow. Email can be sent to multiple email addresses at any point in a process flow. Remember that the server must be configured for sending email.
	Stored Procedure Indicates a saved procedure (other than a data or process flow). The stored procedure can be any FOCUS program that is executed during a process flow.
	Connector Represented as arrows, connector objects specify the logic that governs what is executed next in a process flow. In some cases, the logic is unconditional. In others, the logic is conditional (set either by defaults or by conditions you define). The color of the arrows indicates these relationships: ❑ Black indicates unconditional ❑ Green indicates a success condition ❑ Red indicates a failure condition ❑ Purple indicates a custom-defined condition
	Set variables Set default values for variables used by other objects in the flow.

Simple Process Flow At a Glance

You can use Data Migrator to design process flows ranging from simple, where you use only a few process flow components, to complex, where you take advantage of a wide range of components.

In the example shown in the following image, the process flow includes a data flow that is scheduled to run weekly, on Thursdays, starting at midnight. At the end of each run, an email message is sent to a designated recipient to indicate the successful completion of the data flow processing.



To create a simple process flow:

1. Open a new flow and click the *Process Flow* tab.

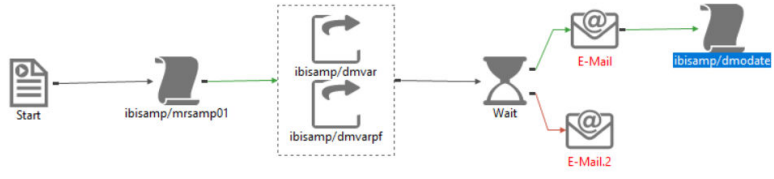
   The Start object appears by default.

2. To add a data flow object, drag the data flow from the navigation pane into the workspace.
3. To add an email object and create a message, ensure that your server is configured for email notification.
4. To add a Schedule, specify when and how often to run the process flow in the Start object properties. For more information, see *Scheduling a Process Flow* on page 406.
5. Connect the objects.
6. Save and run the process flow.
7. View the log file for flow processing details.

Complex Process Flow at a Glance

You can design a complex process flow by combining a wider range of objects in the workspace. You can run procedures before and after executing flows, execute flows concurrently or individually, and send email messages based on the outcome of the execution.

In the process flow shown in the following image, a setup procedure prepares the environment for data extraction. Two data flows are grouped to run simultaneously, followed by a Wait object. When the processing of both data flows is complete, one of two email messages is sent: one if the previous step is successful, the other if the previous step is unsuccessful. After a successful run, a cleanup procedure is executed to prepare for the next scheduled process flow run.



To create a complex process flow:

1. Open a new flow and click the *Process Flow* tab.

   The Start object appears by default.

2. Add a procedure to set up the environment prior to data extraction, and connect it to the Start object.

   This procedure runs at the start of the process flow.

3. Add a Parallel Group object for concurrent processing of two data flows, and connect it to the procedure.

   This enables you to run multiple data flows simultaneously.

   Tip: Right-clicking the Parallel Group line opens a context menu, which allows you to delete the whole group, or select all objects in the group and move them. You can then either delete the frame or add other objects to the group.

4. Add the data flows to the Parallel Group object.

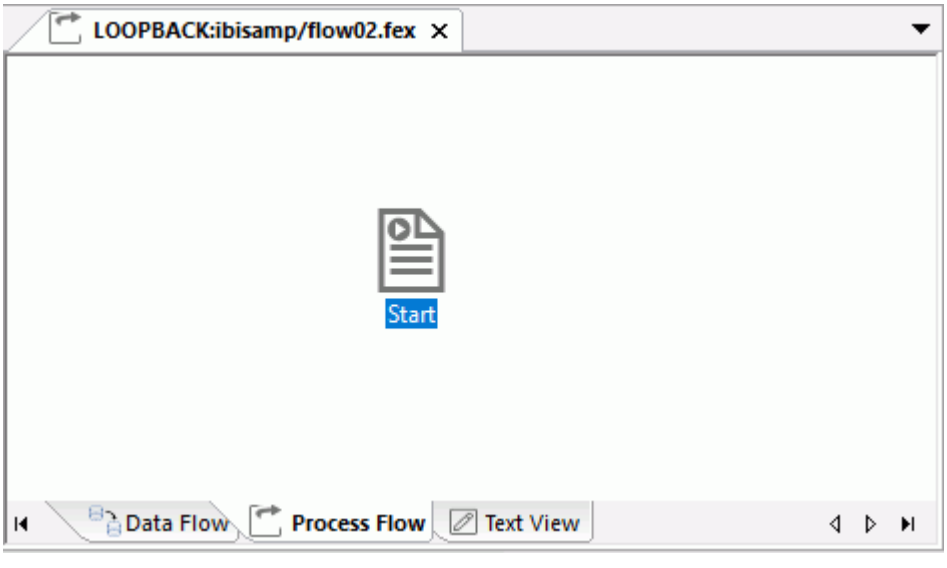
   These data flows run simultaneously.

5. Add a Wait object to ensure that the next step does not begin until all of the data flows in the Parallel group are complete.
6. Connect the Group object to the Wait object.
7. Add another Data Flow object to be executed when the Wait conditions are satisfied.
8. Connect the Data Flow object to the Wait object.
9. Create two email messages and connect both to the Data Flow object, forming two branches in the process flow.
10. Send the messages to appropriate recipients when the data run succeeds or fails.
11. Add a cleanup procedure that runs after the data flows, and that connects to the email object upon successful completion the flow.
12. Open the Start object and specify when and how often to run the flow.
13. Save and run the process flow.
14. View the log file for flow processing.

Creating a Process Flow

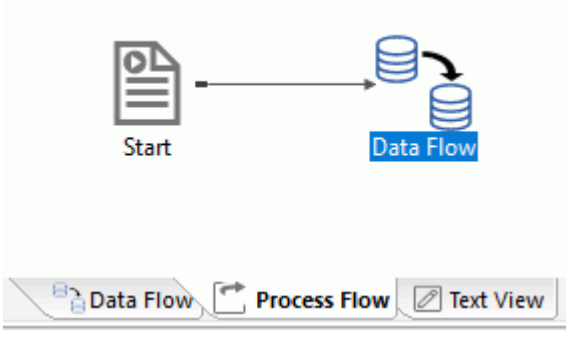
You can create a process flow either by starting a new flow (the preferred method), or by building on an existing data flow (which is suitable for some simple situations). In either case, you add objects to the process flow base, which varies slightly depending upon the method you choose. For more information about these approaches, see *Process Flow Methodology* on page 370.

- If you create a new flow, the process flow workspace initially contains only the Start object, as shown in the following image. You can then begin to add other objects to the flow.



For details, see *How to Create a Reusable Process Flow* on page 377.

- If you build upon an existing data flow, the process flow workspace initially contains the Start object connected to the embedded data flow object, as shown in the following image. You can then add others to the flow.



For details, see *How to Create a Process Flow From a Data Flow* on page 378.

With either method, the Start object can only have one object as output. It can be any object type except for a Wait object, which can only follow a Parallel Group object.

The connector arrow from the Start object to the next object in the flow will always be black (that is, unconditional) and cannot be changed.

Procedure: How to Create a Reusable Process Flow

This procedure describes the recommended modular process flow design methodology described in *Process Flow Methodology* on page 370. To create a reusable process flow:

1. Right-click an application directory in the navigation pane, point to *New*, and then click *Flow*.
2. Click the *Process Flow* tab. A new workspace containing a Start object opens in the workspace.
3. To add objects to the workspace:
   - Drag a data flow, another process flow, or a procedure into the workspace from the navigation pane. You can open another flow within the process flow by right-clicking it and clicking *Open*.
   - Click an object from the Insert group and drag it to the workspace. Then double-click the object to open a properties specification pane. See the detailed topics in this chapter for information on property specifications for each type of object.

4. To connect objects in the workspace, right-click the left object and, holding the mouse button down, drag from the left object to the right object, creating a connecting arrow. Release the mouse button.
5. When you have finished building the process flow, save and name it.
   a. Click the *Save* or *Save As* button from the Quick Access Toolbar. The Save Procedure As dialog box opens.
   b. Name the process flow.
   c. Save it to any application directory in your application path.
   d. Click *OK* to return to the Process Flow workspace.

      Alternatively, you can save and name the process flow at any point during development, and then continue to click *Save* to update the flow as you build it.

Procedure: How to Create a Process Flow From a Data Flow

Use the following steps only when you wish to create a process flow that contains a limited number of objects, and when you do not intend to use the embedded data flow with other process flows.

1. After creating and saving a data flow, click the *Process Flow* tab. The Start object and the embedded Data Flow object are connected by default.
2. To add new objects to the workspace, you can do either or both of the following:
   - ❑ Drag a data flow, another process flow, or a procedure into the workspace from the navigation pane. A process flow can be comprised of data flows, process flows, and/or procedures from any application directory in the search path of the user.
   - ❑ Click an object in the Insert group and drag it into the workspace. Then double-click the object to open a properties specification pane. See the detailed topics in this chapter for information on property specifications for each type of object.
3. To connect objects in the workspace, right-click the left object and, holding the mouse button down, drag from the left object to the right object, creating a connecting arrow. Release the mouse button.

   Tip: If you want to add an object between the Start object and Data Flow object, right-click the connecting arrow and choose *Delete*. Then add the new object. Connect the Start object to the new object, and then connect the new object to the Data Flow object.
4. When you have finished building the process flow, save it under the data flow procedure name. Both the data flow and process flow logic are contained in the same procedure.

   Alternatively, you can save the process flow at any point in the development cycle.

For an illustration of this technique, see *Procedures Shortcut Menu Options* on page 114.

Saving a Process Flow

Process flows are saved on the Server. The first time you save a process flow, you will select an application directory and assign a name to it. After that, you can save it with the same name or assign a new one.

The following restrictions apply to flow names:

- Names can contain only letters, numbers, and underscores.
- Uppercase letters are converted to lowercase when the flow is saved.
- Names can be up to 64 characters long, except on z/OS with PDS deployment, where names are limited to 8 characters.

Procedure: How to Save a Process Flow

1. Click the *Save* button from the Quick Access Toolbar.

   Note: A flow can only be saved once changes have been made to it. If no changes have been made, both the button and Save option in the main menu will be inactive (grey).

   If this is the first time a flow is being saved, the Save Procedure As dialog box will open. See *How to Save a Data Flow with a New Name* on page 182.

2. Check the console window for a message indicating that the flow was saved successfully.

Procedure: How to Save a Process Flow with a New Name

1. Click the *Save As* button from the Quick Access Toolbar.

   The Save Procedure As dialog box opens.

2. From the *Save in:* drop-down menu, select the application directory where you want to save the flow.
3. In the *Name* box, enter the name you wish to use for the flow.

   Click *Save*.

4. Check the console window for a message indicating that the flow was saved successfully.

Adding a Saved Procedure to a Process Flow

You can add any previously saved procedure to the process flow you are currently designing. The procedure may be:

- **A data flow.** For details about creating and saving data flows, see *Designing a Data Flow* on page 171.
- **Another process flow.** The current chapter describes all aspects of this task.
- **A procedure.** For details, see *How to Create a Procedure* on page 381.

Procedure: How to Add a Flow to the Process Flow

1. Right-click an application directory in the navigation pane, point to *New*, and then click *Flow*.
2. Click the *Process Flow* tab. A new workspace containing a Start object opens in the right pane.
3. From the navigation pane, select a flow and drag it into the process flow workspace. The flow may be a data flow or another process flow.
4. Right-click the object to the left of the flow and, holding the mouse button down, drag from the left to the right object, creating a connecting arrow. Release the mouse button.

   The procedure will be executed within the process flow at the point where it is encountered.
5. Click the *Save* button to update the process flow.

Procedure: How to Add a Procedure to the Process Flow

1. Right-click an application directory in the navigation pane, point to *New*, and then click *Flow*.
2. Click the *Process Flow* tab. A new workspace containing a Start object opens in the right pane.
3. From the navigation pane, select a procedure and drag it into the process flow workspace. For related information, see *Procedures and Flows* on page 381.
4. Right-click the object to the left of the procedure and, holding the mouse button down, drag from the left object to the right object, creating a connecting arrow. Release the mouse button.

   The procedure will be executed within the process flow at the point where it is encountered.
5. Click the *Save* button to update the process flow.

Reference: Procedures and Flows

In the course of developing a flow (either a data flow or a process flow), you may need to perform a variety of specialized tasks, from setting up the environment parameters and variables, to copying files. The DMC provides an editing window in which you can type your 4GL code or SQL. When you save the file, it is stored in an application directory, identified by the icon .

Then you can:

- Add it as an object in a process flow.

  In fact, you can create a process flow composed of procedures and other objects, such as email messages, without any data flows. See *Executing a Process Flow That Contains Only Procedures* on page 393.

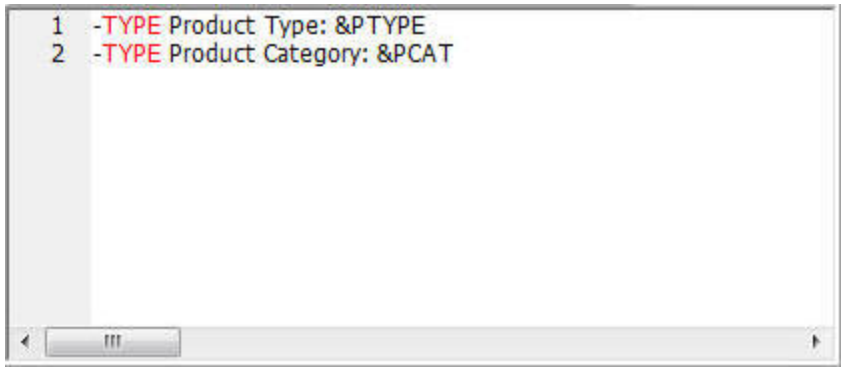
- Use it in the design and testing of a data flow. For details, see *Using Variables in a Flow* on page 314.

Procedure: How to Create a Procedure

To create a procedure:

1. Right-click an application directory in the navigation pane, point to *New*, and then click *Procedure*.
2. A text editing window opens. Type the procedure code, which may be either FOCUS or SQL.

   For example, create a procedure named PRODPARMS that has two parameters, &PTYPE and &CAT, as shown in the following image.



   You are responsible for the accuracy of the syntax you enter. However, any errors that are generated will be displayed in the Server Messages tab below the workspace when you execute the procedure.

3. Click the *Save* or *Save As* button from the Quick Access Toolbar. Enter a procedure name in the Save Procedure As dialog box.

   The procedure is listed in the navigation pane under an application directory, where it is preceded by the Procedure icon.

   From here you can drag it into a process flow at any point. For details, see *Adding a Saved Procedure to a Process Flow* on page 380.

Assigning and Using Variables in a Process Flow

Variables allow you to control the execution of a process flow by substituting values at run time. A Set Variables object lets you set the value for a variable to be used in other objects. These variables store a string of text or numbers that can be used in the process flow, passed as parameter to a data flow, or used in a procedure.

Some common uses of variables are in a SELECT statement to add a constant value to an answer set or in a WHERE condition to restrict the rows selected. Another use of a variable is to assign a name for a target table.

Procedure: How to Set a Value for a Variable

1. Right-click an application directory, point to *New*, and then click *Flow*.

   A data flow window opens.

2. Click the *Process Flow* tab to switch to that view.
3. On the Flow tab, in the Insert group, drag the Set Variables button onto the workspace.
4. Right-click the *Start* icon, drag the cursor to the Set Variables icon, and release.

   The workspace should resemble the following image.



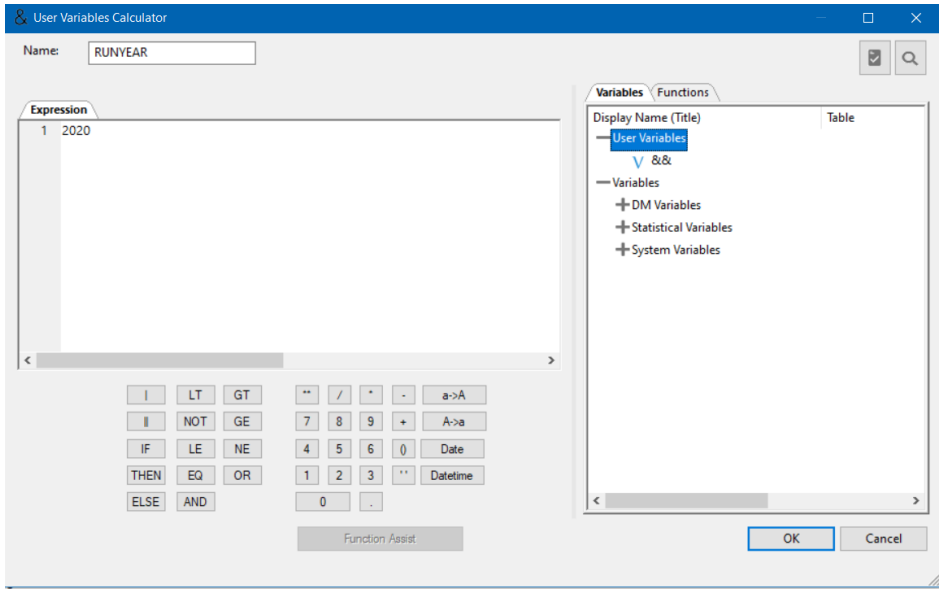
5. Right-click the *Set Variables* object and click *Variables Editor*.

   The Set Variables Editor opens.

6. Click the Add SET Parameter button.

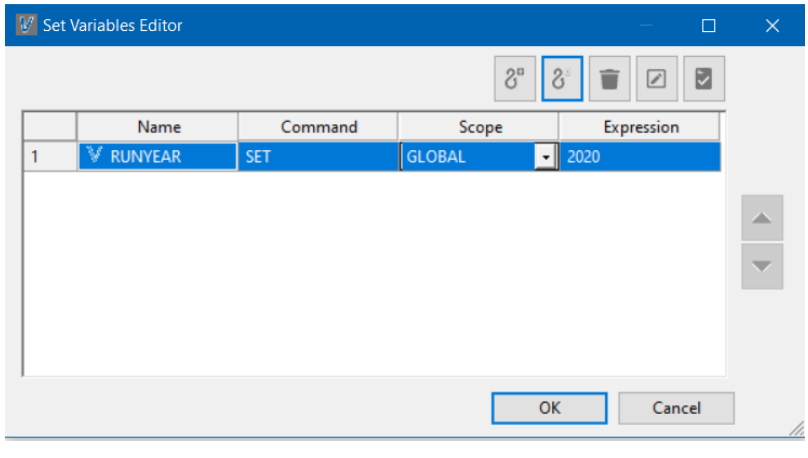
   The User Variables Calculator opens.

7. Enter a name and value for the variable, as shown in the following image.



8. Click *OK*.

   The Set Variables Editor now displays the details for the parameter, as shown in the following image.



9. Click *OK* to close the window.

The global variable &&RUNYEAR can now be used in any procedure added to the process flow after the Set Variables object. It can also be used in an embedded (purple) data flow, or in a data flow that is added to the process flow, provided that the Run as procedure option is selected in the properties page for that data flow.

Reference: **Set Variables Editor Toolbar**

The Set Variables Editor allows you to assign values to variables.

The Set Variables Editor toolbar contains the following buttons:

Button	Function
	**Add DEFAULT Parameter** Adds a DEFAULT parameter.
	**Add SET Parameter** Opens the User Variables Calculator to assign a value to a parameter.
	**Delete Parameter** Deletes the currently selected parameter.
	**Edit Selected Parameter** Opens the User Variables Calculator for the currently selected parameter.
	**Test Variables** Tests the variable assignments.

The Set Variables Editor also contains the following input areas:

Name

Name of the variable.

Command

Provides a drop-down menu with the options of SET or DEFAULT.

Scope

Provides a drop-down menu with the options of LOCAL or GLOBAL.

Expression

Value to be assigned to the variable.

Reference: Variable Names, Types, and Assignments

A variable name may be any combination of the characters A through Z, 0 through 9, and the underscore. The first character of the name should be a letter. Variable names are preceded by one or two ampersands, depending on the type.

There are two types of variables that can be assigned values in a Process Flow.

LOCAL

Assigns a value for a variable that is used throughout the current flow. It can be used in the current process flow in conditions, email objects, and as parameters to data flows and process flows.

GLOBAL

GLOBAL assigns a value for a variable that is used throughout the session. In addition to the places where a local variable can be used, global variables can also be used in the current process flow by any procedures it invokes, as well as any data or process flow that it invokes, provided that *Run as procedure* is selected in its properties.

There are two ways of assigning a value for a variable within a process flow:

DEFAULT

Supplies an initial (default) value for a variable that had no value before the command was processed. It ensures that values are passed to the variable, whether or not they are provided elsewhere. If the variable already has a value, the command is ignored.

SET

Assigns a value for a variable, which overrides any previously set values.

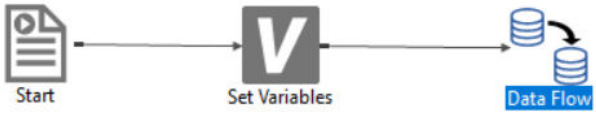
Procedure: How to Set a Global Variable

To set a global variable and its value:

1. Open a data flow.
2. Click the *Process Flow* tab.

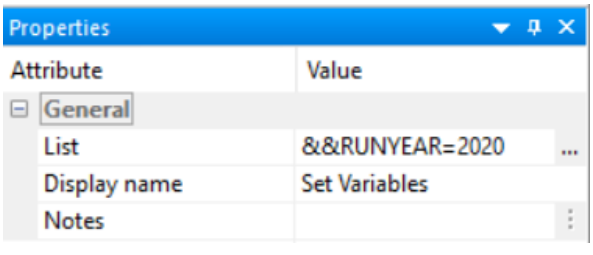
   The Process Flow workspace opens.
3. Right-click the line joining Start and Data flow and click *Delete*.
4. On the Flow tab, in the Insert group, drag the Set Variables button into the workspace between Start and Data Flow.

5. Right-click Start, drag a joining line to Set Variables, and release. Do the same between Set Variables and Data Flow. Your workflow should look like the following image.



6. Double-click the Set Variables object. The Set Variables Editor opens.
7. Click the *Add SET Parameter* button to open the User Variables calculator. In the Expression box, enter a name for the variable and a value. Click *OK*.

   The Set Variables Editor now shows the name and value entered.
8. Click *OK*. The name and value appear on the Properties pane, as shown in the following image.



Procedure: How to Pass Parameters in a Process Flow

When you are using a procedure in a process flow, you can change the value of its parameters when the flow runs.

From the Process Flow tab:

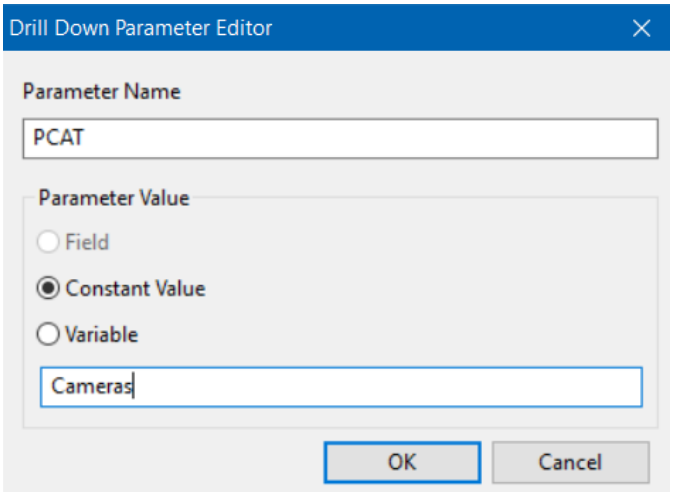
1. Double-click the procedure.

   or

   Right-click the procedure and click *Properties*.

   The Properties pane opens.
2. Select the Parameters ellipsis button to open the Parameter Editor. Then, select the Add Parameter  button to open the Drill Down Parameter Editor.

Enter *PCAT* in the Parameter Name field. Leave Constant Value selected and enter *Cameras* as a parameter value, as shown in the following image.



3. Click *OK*.

   The value appears in the Parameter Editor. Repeat the process for the variable PTYPE, using a parameter value of Digital.

   The parameters can also be entered directly in the Properties pane by following the guidelines below:

   - ❏ Do not include the ampersand in the variable name.
   - ❏ Use single quotes around character and date valued variables.
   - ❏ Separate multiple variables with commas.

4. When done, click *OK* to close the Parameter Editor.

   The parameters you entered appear in the Properties page.

5. Save and submit the flow.

Reference: Data Migrator Variables

Most Data Migrator variables start with &&CM__ (double underscores). They set the following variables in each flow:

&&CM__AUTHOR

Is the user ID that originally saved the flow.

Note: &&CM__AUTHOR is the user ID that originally saved the flow, which is not always the user ID that ran the flow. To find the user ID that ran the flow, assign a value of

```
GETUSER('A32')
```

to a variable in the set variables object on the process flow tab.

&&CM__REQUEST

Is the name of the flow.

&&CM__TARGET

Is the name of the target table for the data flow. Note that for multi-table requests, the variable CM__TARGET has a number appended to the name.

When you run a flow, these variables are automatically set and can be used in a procedure that follows. For example, you can display them in logs, embed them in reports, and use them in email messages. You can reference these variables in procedures that run before or after data flow execution.

Reference: Control Variables

Control variables are typically used in process flows.

&&KILL_RPC

Shuts down all flow processing if the value is set to Y. The flow immediately branches to the end of its logic, skipping all code that follows the procedure where this setting is used. Any subsequent data flows or procedures are not executed. This may be useful in cases where a site-dependent variable (such as a program result or special file allocation) fails, and the entire extract is to be stopped. The default is N.

To reset, type:

```
-SET &&KILL_RPC = 'Y';
```

Reference: Statistical Variables

Data Migrator supports the following statistical variables that can be referenced in procedures. Statistical variables are primarily used in process flows.

&ACCEPTS

Is the number of transactions accepted.

&CHNGD

Is the number of rows updated.

&DBMSERR

Is the number of rows rejected due to database constraint violations, or for fixed format file targets, format conversion errors.

&DELTD

Is the number of rows deleted.

&DUPLS

Is the number of rows rejected as a result of duplicate values in the data source.

&FOCERRNUM

Is the last error number displayed. If the last command did not generate an error, the value of &FOCERRNUM is 0.

&FORMAT

Is the number of rows rejected as a result of a format error.

&INPUT

Is the number of rows added to the data source.

&INVALID

Is the number of rows rejected as a result of a VALIDATE condition.

&NOMATCH

Is the number of rows rejected as a result of not matching a value in the data source.

&REJECTS

Is the number of rows rejected for reasons other than the ones specifically tracked by other statistical variables.

&TRANS

Is the number of rows processed.

&LINES

Is the number of lines returned in the last answer set.

&READS

Is the physical number of reads from an external file.

&RECORDS

Is the number of records retrieved in the last answer set.

Reference: Slowly Changing Dimension Variables

Slowly Changing Dimension variables are used to control flows that have their target properties Load Type set to Slowly Changing Dimensions. They allow overriding of the default begin and end dates when rows with Type II changes are updated and inserted.

&&CM__SCDATE

Overrides the current date as the begin date and end date for Type II changes. This setting is typically used when a run was missed, and a rerun should be run as though it were run on the earlier date. For example, the following could be used so that an effective date, June 1, 2020, is used by including it in a procedure that is run before the data flow.

```
-SET &&CM__SCDATE = 20200601;
```

&&CM__SCDEIDT

Overrides the current date as the end date for Type II changes for inactive rows. This setting is typically used so that the day prior to the run date becomes the end date. For example, if today is June 1, 2020, then either of the following could be included in a procedure run before the data flow, so that an end date of May 31, 2020 is used.

```
-SET &&CM__SCDEIDT = 2020531;
```

or

```
-SET &&CM__SCDEIDT = AYMD(&YYMD, -1, 'I8YYMD');
```

&&CM__SCDEADT

Overrides a NULL as the end date to use for Type II changes for active rows. This setting is typically used with a date far in the future. For example, the following could be used in a procedure before the data flow, so that January 1, 3000 is used as the end date.

```
-SET &&CM__SCDEADT = 30000101;
```

Maintaining Continuous Processing in a Single Agent

By default, each time you run a flow, either standalone or as a component in a process flow, a new agent is started. However, you may want to run a flow in the agent that is currently running, rather than start a new agent. Suppose, for example, that you run a procedure that issues a FILEDEF, a SET command, or that creates a global (&&) variable, and you want to reference the setting or variable in the flow that follows the procedure. Since a flow that runs in a new agent will not recognize or be able to use these settings or variables, you need to maintain continuous processing in the same agent.

Data Migrator supports the runtime requirement. To ensure that the continuous connection is maintained at runtime, you can request that a flow that creates input required by another flow be marked to *Run as Stored Procedure*. See *How to Maintain Continuous Processing in a Single Agent* on page 391.

Procedure: How to Maintain Continuous Processing in a Single Agent

1. From the navigation pane, select a flow, and drag it into the process flow workspace. The flow may be a data flow or another process flow.
2. Right-click the object to the left of the flow and, holding the mouse button down, drag from the left object to the right object, creating a connecting arrow. Release the mouse button.

   The procedure will be executed within the process flow at the point where it is encountered.
3. Double-click the flow object to open the Properties pane.
4. Select the *Run as Stored Procedure* check box to execute the flow in the running agent. For details, see *Properties Pane for Flows and Procedures* on page 391.
5. Close the window.
6. Click the *Save* button to update the process flow.

Tip: If you switch to Text View and examine the underlying execution code for a flow, you will see EX CMASAP as the flow name. CMASAP triggers a connection to a new agent. If you select *Run as Stored Procedure* in the Properties pane, the code shows EX flow name and keeps the same agent open for processing.

Reference: Properties Pane for Flows and Procedures

To access the Properties pane, double-click a flow object or procedure.

The Properties pane has the following fields and options:

Flow or Procedure Name

Displays the name of both the object and the directory in which the flow or procedure is stored.

If you drag a Flow icon from the toolbar, you can enter:

- The name of an existing procedure and its application directory.
- The name of a new procedure that you must build and save.

Note: If you have already created and saved the procedure, the application directory and procedure name appear when you open the window.

Get Parameters using Synonym

Select this check box to obtain parameters using a synonym, instead of entering them in the Properties pane.

When this option is used, a variable is automatically generated with the name of each field in the selected synonym. If the associated data source of the synonym has more than one row, the flow is submitted once for each row, with values for the variables obtained from each row.

For an example of using this option, see *Iterating Through Multiple Tables* on page 612. If you select this option, the following fields are available:

- **File Name.** Enter or select the application directory and name of a synonym for Data Migrator to use for obtaining parameters. This option only appears when the Get Parameters using Synonym option is selected.
- **Parallel Run.** When this option is selected and the associated data source of the synonym has more than one row, flows are submitted asynchronously. This option only appears when the Get Parameters using Synonym option is selected and is only available for flows.

Parameters

Indicates any parameters that should be used when the flow or procedure is run.

Clicking the ellipsis opens the Parameter Editor for setting the parameter values. For more information, see *How to Pass Parameters in a Process Flow* on page 386.

Run as Stored Procedure

Select this check box if you wish to run the flow in the agent that is currently running, rather than by connecting to a new agent. This option is only available for flows. For example, you would use this method to maintain continuous processing:

- If one data flow creates a temporary fixed format file that is used as an input to the next flow.
- If a procedure creates a global variable, or issues a FILEDEF or SET command that you need to reference in the new flows.

Do not allow multiple instances

Select this option to prevent this flow from running if another instance is already running. This option is only available for flows.

Display Name

Displays the name of the flow or procedure.

Notes

Add notes describing the purpose of the flow or procedure.

Executing a Process Flow That Contains Only Procedures

You can create a process flow that contains one or more procedures and other objects, but no data flows. This may be useful for testing purposes, or for a particular warehouse scenario where the Data Migrator Scheduler is being used, but the only production code is the logic within the procedure.

For example, you can create and run a procedure that connects you to a DBMS, or that produces a report from a data file or a Data Migrator table, such as the Data Migrator log.

Procedure: How to Execute a Process Flow That Contains Only Procedures

1. In the DMC, right-click an application directory in the navigation pane, point to *New*, and then click *Flow*.
2. Click the *Process Flow* tab.
3. Drag a procedure from the navigation pane into the process flow workspace, following the Start object.
4. Connect the objects.
5. Optionally, add other procedures to the process flow.
6. Save the procedure.
7. On the Flow tab, in the Run group, click *Run*, and then click *Run* from the shortcut menu.

   or

   Right-click the saved process flow in the navigation pane and choose *Run* from the menu.

Output is displayed in the Console Log below the workspace.

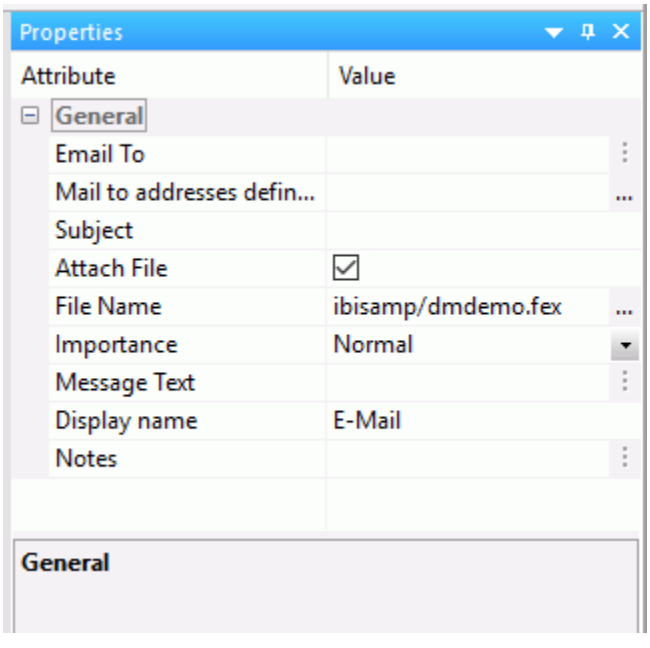
Adding Email Messages to a Process Flow

Before you create email messages for a process flow, you must complete or verify your email notification setup in the Web Console. For details about accessing the E-Mail Notification tab and configuring SMTP services, see *Configuring TIBCO WebFOCUS Reporting Server Operations* on page 65.

Once SMTP services are properly configured, you can construct your messages within the process flow. You can include email messages at any point in the flow.

Procedure: How to Add an Email Object to the Process Flow

1. On the Flow tab, in the Insert group, drag the *E-Mail* icon from the ribbon into the process flow workspace.
2. Double-click the *E-mail* object to open the Properties pane, as shown in the following image.



3. In the Email To box, type the email addresses of one or more recipients. Separate multiple addresses with semi-colons (;).
4. In the Subject box, type a subject.

   Note: Optionally, select a file to attach. Files of any type can be attached to the email object.

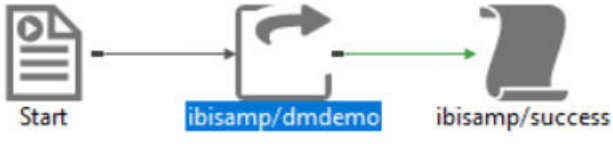
5. Optionally, select the Attach File checkbox to add an attachment. Once selected, a File Name box becomes available. Here, you can either type in the file name, or use the ellipses button to browse for the file. Any file type or extension is supported.
6. In the Message Text box, type your message.
7. Close the pane.
8. Click the *Save* button to update the process flow.

Note: You can also add email properties from the Web Console.

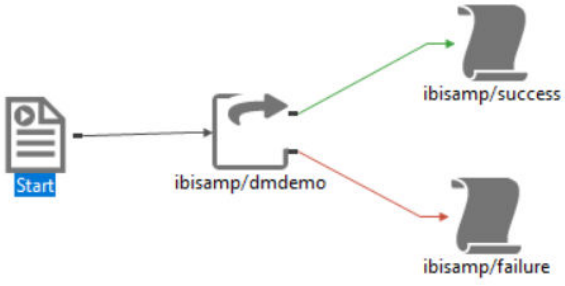
Setting Connection Logic

When you connect one object to another, Data Migrator draws an arrow between them. These arrows, or *connector objects*, specify the logic that governs what is executed next in a process flow. The connection logic may be unconditional or conditional, represented graphically by colored arrows. Conditional logic is set by default. However, you can edit the connection to change the nature of the condition or to make it unconditional.

- Black arrows indicate unconditional connection logic. This means that the next object will be processed immediately after its predecessor. For some pairs of objects, unconditional logic is required. The connections between the following types of objects and their successors are unconditional by default, and cannot be changed:
  - Start connects to any other object type
  - Parallel Group connects to Wait
- Green arrows indicate that the next object will be processed if the previous one runs successfully. This is the default when you draw an arrow from one object to the next in a single path, as shown in the following image.



- Red arrows indicate that the next object will be processed if the previous one fails to run properly. This is the default for the second arrow when you draw two arrows from the same object to other objects, forming two branches in the flow. Since the first arrow (the green one) means run when the flow is successful, it follows that the second arrow (the red one) means run when the flow fails, as shown in the following image.



- Purple arrows indicate that the next object will be run based on a custom condition you define. You can use statistical or global variables (&&variables) in custom conditions to test specific conditions, return codes, or values based upon which branching will occur. For example, you might want the flow to branch differently based a particular error message number. For related information, see *Branching in a Process Flow* on page 401.

  Note: When multiple conditional arrows are used, the first object where the condition is met is run.

The connections between the following objects are conditional by default. You can change the conditions or make them unconditional:

- Email connects to any other object type.
- Flow connects to Flow, Parallel Group, Procedure, Email. (The conditions you can apply are: OK, FAILURE, and a Custom condition.)
- Procedure connects to Flow, Parallel Group, Procedure, Email. (The conditions you can apply are: OK, FAILURE, and a Custom condition.)
- Wait connects to Flow, Parallel Group, Procedure, Email. (The conditions you can apply are: Wait satisfied and Timeout.)

Procedure: How to Edit a Connection Condition

Follow these steps to edit a connection condition between a flow or a procedure and one of the following object types: a flow, a Parallel Group, a Procedure, or an Email.

1. Drag the required objects into the process flow workspace.
2. Connect the objects.
3. To view or edit the connection condition, double-click the arrow connecting the two objects. The Condition dialog box opens.

   For a single connection, OK is the default. For a second connection, Failed is the default.
4. Click a radio button to change the condition.
5. If you wish to specify a Custom condition, click *Custom*. The Edit Custom Focus Condition box becomes active.

   Type the code for the condition you want to apply. Typically, you use a global (&&) variable, Data Migrator variable, or statistical variable in your custom conditions to test specific conditions, return codes, or values based upon which branching will occur.
6. Click *OK* to close the condition box.
7. Click the *Save* button to update the process flow.

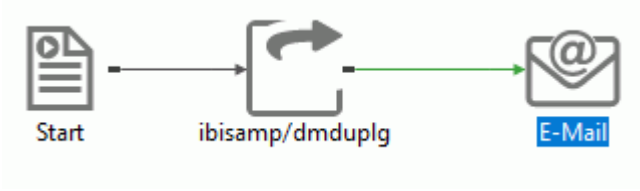
Example: Creating a Custom Connection Condition

You can set a custom condition for a connection between objects in a process flow. In this case, a connection will become active if duplicate rows are rejected while a data flow is run. The number of duplicates will be included in an email that is sent if the condition is met.

Note: Your Server must be configured for email notification to complete this example. For details, see *Configuring TIBCO WebFOCUS Reporting Server Operations* on page 65.

1. In the DMC, right-click an application directory in the navigation pane, point to *New*, and then click *Flow*.
2. Click the *Process Flow* tab.
3. Drag the *DMDUPLG* data flow object from the ibisamp application directory in the navigation pane into the workspace, to the right of the Start object.
4. Create a connection by right-clicking the *Start* object and dragging to the *DMDUPLG* data flow object.
5. On the Flow tab, in the Insert group, drag an email object into the workspace, to the right of the DMDUPLG data flow object.
6. Create a connection by right-clicking the *DMDUPLG* data flow object and dragging to the email object.

The workspace should appear, as shown in the following image:



7. Double-click the arrow between the *DMDUPLG* data flow object and the email object.

   or

   Right-click the connection arrow and click *Edit Condition*.

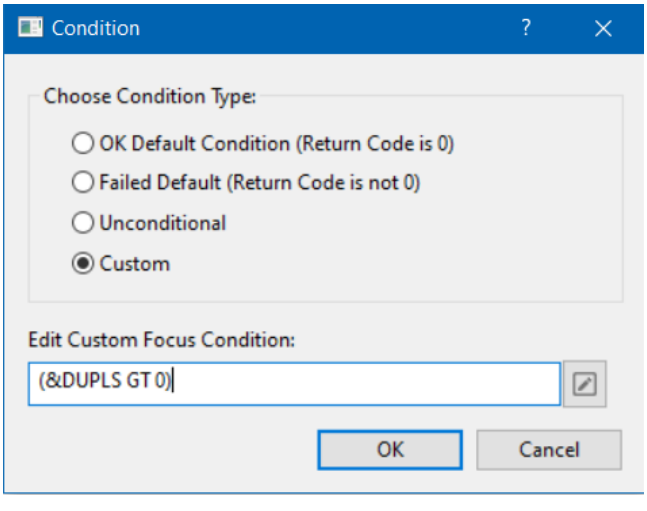
   The Condition dialog box opens.

8. Click the *Custom* radio button.

   The Edit Custom Focus Condition field becomes active.

9. Type the following in the Edit Custom Focus Condition box and click *OK*, as shown in the following image.

   ```
   (&DUPLS GT 0)
   ```



   Note: Ensure the Data Flow that was added to the Process Flow has the *Run as Stored Procedure* setting enabled in the Properties pane. This will maintain the continuous connection, and will enable the reference to the variables at run time.

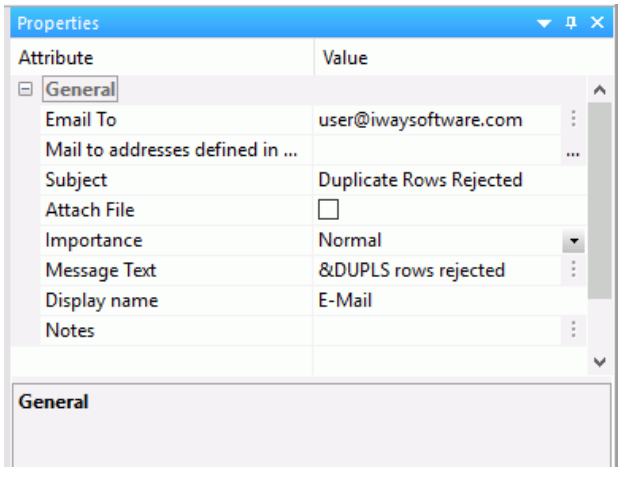
10. Right-click the email object and click *Properties*.

    The Properties pane opens.

11. In the Email To box, type an email address. In the Subject box, type *Duplicate Rows Rejected*.

12. Type the following in the Message Text box, as shown in the following image.

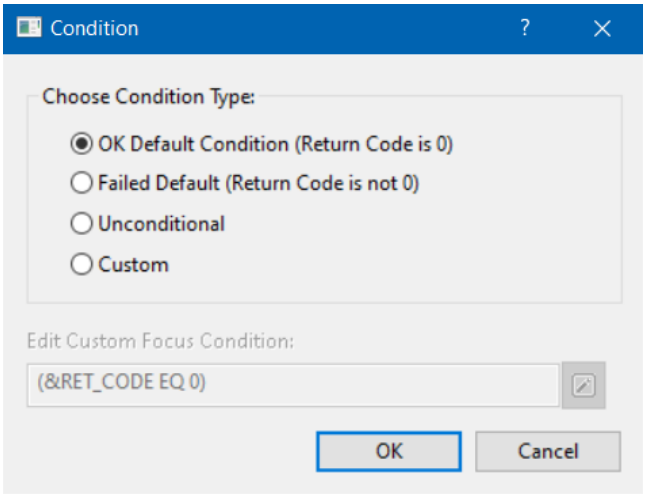
```
&DUPLS rows rejected
```



This connection will become active if duplicate rows are rejected while a data flow is run, and an email notification will be sent with the number of duplicate rows.

Reference: Condition Dialog Box

To access the Condition dialog box shown in the following image, from the process flow workspace, double-click a connection arrow between objects, or right-click a connection arrow and click *Edit Condition*.



The Condition dialog box has the following fields and options:

OK Default Condition (Return Code is 0)

Executes the connection if the prior object executed successfully.

Failed Default (Return Code is 0)

Executes the connection if the prior object failed to execute.

Unconditional

Executes the connection regardless of the success or failure of the prior object. This option is only available when there is only one connection.

Custom

Allows you to set a custom condition for the connector.

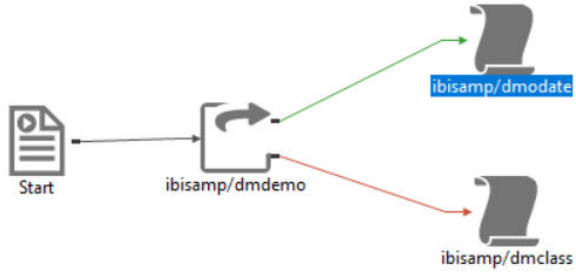
Edit Custom FOCUS Condition

The code for a custom condition. Enter a condition using FOCUS Dialogue Manager syntax that evaluates to true or false.

Branching in a Process Flow

The ability to set connection logic enables you to create branches within a process flow. You simply connect multiple objects to the object from which you want the branching to begin.

The branching process flow shown in the following image is built in *Exercise 2: Creating the Process Flow* on page 445. It is strongly suggested that you complete these exercises in order to create the branches shown here. As you do so, you will become familiar with many aspects of data and process flow design in the DMC.



Procedure: How to Create Branches in a Process Flow

1. Drag an object into the workspace following the object from which you want the branching to begin.
2. To build the first branch, connect the object at the starting point for the branches to the object you just dragged in. A green arrow is created by default.
3. To build the second branch, drag another object into the workspace and connect it to the object at the start of the branch. A red arrow is created by default.
4. If you wish to change the connection conditions for any branch, double-click the arrow and edit the condition in the Conditions dialog box. Then click *OK*.
5. Click the *Save* button to update the process flow.

Note: You can also create branches using custom conditions. Note, however, that while you can use multiple custom conditions to specify different branches, you should not use custom conditions with OK and FAILED conditions.

When multiple conditional arrows are used, the first object where the condition is met is run.

Grouping and Controlling Simultaneous Execution of Flows

Often you need to define a process flow in which multiple steps occur simultaneously and where other steps cannot begin until all of the prior steps have been completed.

For example, suppose that you are creating an employee information data mart as a data target. However, the data mart requires data from two other independent data flows, one that loads data from a personnel database and one that loads data from an accounting database. You need to make sure that the data flow for the data mart does not run until the two feeder data flows have completed their runs.

By grouping the feeder data flows, you can enforce their simultaneous execution. Then, by adding a Wait object, you can ensure that the third data flow, the employee data mart, does not run until the other two have loaded the required data into their respective targets. This technique eliminates the need to estimate what time the first two flows will end in order to schedule the third flow to run at that time.

You can also pass parameters to the procedures in a parallel group using the Properties pane for the flow by entering a value into the Parameters box.

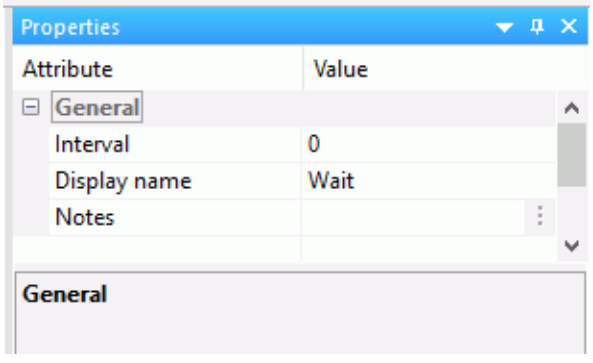
Note: The Scheduler must be running to execute process flows that require simultaneous execution or wait conditions. If the Scheduler is not running, the flow will not run, and a message will appear in the log.

Procedure: How to Execute a Group of Procedures and Wait

1. Right-click an application directory in the navigation pane, point to *New*, and then click *Flow*.
2. Click the *Process Flow* tab. A new workspace containing a Start object opens in the right pane.
3. On the Flow tab, in the Insert group, click *Parallel Group* and drag it into the workspace, where it appears as an empty box with a border of dashes.
4. Right-click the previous object in the workspace and draw an arrow connecting it to the Parallel group object.
5. Drag a flow (it can be a data flow or another process flow) from the navigation pane into the Parallel Group object.
6. Drag another flow from the navigation pane into the Parallel Group object, placing it below the previous object.

   Note: You can group as many flows or procedures as you require in a Parallel Group object.
7. On the Flow tab, in the Insert group, click *Wait*, and drag it into the workspace to the right of the Parallel Group object.
8. Right-click the Parallel Group object, which now contains the multiple flows, and draw an arrow connecting it to the Wait object. Note that this is an unconditional connection.

9. By default, the Wait object simply waits until the flows in the Parallel Group object complete their runs. If you want to limit the wait time, double-click the Wait object to open the Wait attribute pane, as shown in the following image.



   In the Interval text box, specify the maximum number of seconds to wait before executing the next component in the flow.

10. From the navigation pane or the Insert group, drag the next object into the workspace, to the right of the Wait object.
11. Connect the *Wait* object to the object that follows it in the flow. By default, the processing will continue to the next object when the default condition is satisfied.
12. If you wish to change the default condition, double-click the connecting arrow to open the Wait Condition Type window. See *Wait Condition Type Dialog Box* on page 405.

    Change the condition to *Timeout or Failure* and click *OK*.

13. Click the *Save* button to update the process flow.

Procedure: How to Edit a Wait Connection Condition

1. Drag a Flow, Procedure, or E-mail object into the process flow workspace, following a Wait object.
2. Connect the objects.
3. To view or edit the connection condition, double-click the arrow connecting the objects. The Wait Condition Type window opens.

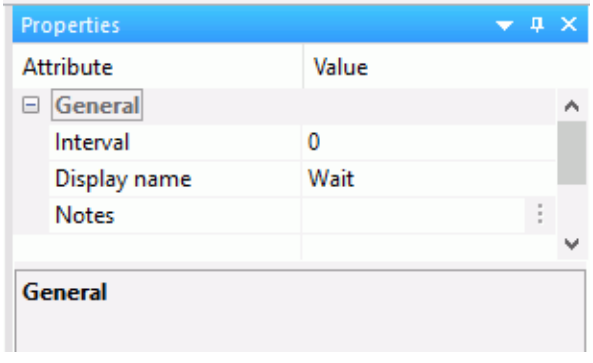
   For a single connection, the default condition is Wait satisfied, represented by a green arrow.

   For a second connection, the default condition is Timeout or Failure, which shows that either the Wait is not satisfied OR one of the procedures in the group failed. This is represented as a red arrow.

4. Click an option button to change the condition, and then click *OK* to close the window.
5. Click the *Save* button to update the process flow.

Reference: Wait Attribute Pane

To access the Wait attribute pane shown in the following image, from the process flow workspace, double-click a Wait object.



The Properties pane for a Wait object has the following fields and options:

Interval

Number of seconds to wait for a parallel group to finish executing. The default value is 0.

An interval of 100 causes the process flow to wait up to 100 seconds or until all flows in the group have been completed.

Display name

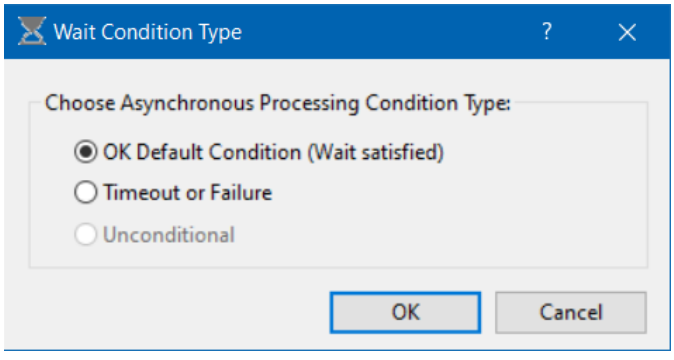
The alternative display name of the object.

Notes

An annotation describing the object.

Reference: Wait Condition Type Dialog Box

To access the Wait Condition Type dialog box shown in the following image, from the process flow workspace, double-click the arrow coming from a Wait object, or right-click it and select *Edit Condition*.



The Wait Condition Type dialog box has the following fields and options:

OK Default Condition (Wait satisfied)

If the grouped objects complete their runs successfully within the set time limit, the wait condition is satisfied and the next upon-successful-completion step in the flow runs.

Timeout or Failure

If the grouped objects do not complete their runs successfully within the set time limit, or one of the grouped objects fails, the wait condition is not satisfied and the next Timeout or Failure step in the flow is run.

Unconditional

The next steps run regardless of any conditions.

Running a Process Flow

There are two ways to run a process flow from the DMC: Run and Submit.

Run will run the process flow immediately. When you run a flow, processing information is displayed in the Console Log.

Submit will submit the process flow to the scheduler to run. When you submit a flow, the processing information is recorded in the Data Migrator Log and Statistics tables. Email messages are also sent if the flow properties are set to send email messages at the start or end of a job.

Note: You can also schedule the flow based on a variety of conditions. For information about scheduling options, see *Scheduling a Process Flow* on page 406.

Procedure: How to Run a Process Flow

There are two ways in which you can run a process flow:

- On the Flow tab, in the Run group, click *Run*, and then click *Run*.
- Right-click a process flow in the navigation pane. Select one of the run options available from the menu.

For more information regarding the run options, see *Procedures Shortcut Menu Options* on page 114.

Scheduling a Process Flow

Data Migrator provides several ways to run data and process flows. In addition to executing flows immediately, you can use the Data Migrator internal scheduler to run requests off-hours, on a recurring basis, or on specified days of the week or month.

Alternatively, you can schedule or execute a flow from an application other than Data Migrator. For details, see *Event-Based Scheduling* on page 681.

You can also disable a scheduler without removing the Schedule from the flow.

You can schedule a process flow to run in the following ways:

- **Run Once.** Runs your flow once.
- **Recurring.** Repeats your flow at an interval that you define.
- **Multi-Day.** Runs your flow on certain days of the month.
- **Run when server starts.** Runs your flow every time the Server starts.

When the scheduler checks, it executes all flows scheduled to run in the period of time since the scheduler last checked.

To access Data Migrator Scheduler parameters:

1. Expand the *Workspace* folder.
2. Expand the *Special Services and Listeners* folder, right-click *SCHEDULER*, and click *Properties*.

For scheduled flows to run, the scheduler must be running. On the Scheduler Configuration dialog box, the *sched_autostart* drop-down menu must be set to yes.

Note: You can also add or change schedule properties from the Web Console.

Procedure: How to Schedule a Process Flow to Run Once

1. Right-click the *Start* object in the Process Flow tab and click *Properties*.

   The Properties pane opens.

2. Select *Active* from the Schedule Status drop-down menu.
3. Select *Run Once* from the Schedule Type drop-down menu.
4. Select a start date in the Start Date field by typing the date in the month/day/year areas, or by clicking the down-arrow to open a calendar in which you can choose a date. The default is the current day.
5. Select a start time in the Start Time field by typing values directly into the hour/minute/seconds/AM or PM areas, or by highlighting an area and using the Up and Down Arrow keys to select new values. The default is the current time.
6. Close the pane.
7. Click the *Save* button to update the process flow.

Procedure: How to Schedule a Process Flow to Run on a Recurring Interval

1. Right-click the *Start* object in the Process Flow tab, and click *Properties*.

   The Properties pane opens.

2. Select *Active* from the Schedule Status drop-down menu.
3. Select *Recurring* from the Schedule Type drop-down menu.
4. Select an interval type (such as daily, monthly) from the Interval Type drop-down menu.
5. Specify an interval number in the Interval Value field to indicate the frequency of execution. For example, if you select Weekly as your Interval Type, and 2 as your Interval Value, the flow will be executed every two weeks.
6. Select a start date in the Start Date field by typing the date in the month/day/year areas, or by clicking the down-arrow to open a calendar from which you can choose a date. The default is the current day.
7. Select a start time in the Start Time field by typing values directly into the hour/minute/seconds/AM or PM areas, or by highlighting an area and using the Up and Down Arrow keys to select new values. The default is the current time.
8. Optionally, select an end date and time by selecting the Specify End Date check box. Specify the end date by typing the date in the month/day/year areas, or by clicking the down-arrow to open a calendar from which you can choose a date. Specify the time by typing values directly into the hour/minute/seconds/AM or PM areas, or by highlighting an area and using the Up and Down Arrow keys to select new values.

9. Optionally, specify special date and time ranges. You can configure the scheduler to run on specific days of the week or month, on the last day of the month, or specify the beginning and ending times of the daily time span.
10. Close the pane.
11. Click the *Save* button to update the process flow.

Tip: If you wish to stop a query from executing regularly, you do not need to change the schedule. Instead, set Schedule Status to *Inactive* in the Schedule attribute pane.

Procedure: How to Schedule a Process Flow to Run at Multi-Day Intervals

1. Right-click the *Start* object in the Process Flow tab and click *Properties*.

   The Properties pane opens.
2. Select *Active* from the Schedule Status drop-down menu.
3. Select *Multi-Day* from the Schedule Type drop-down menu.
4. Select a start date in the Start Date field by typing the date in the month/day/year areas, or by clicking the down-arrow to open a calendar from which you can choose a date. The default is the current day.
5. Select a start time in the Start Time field by typing values directly into the hour/minute/seconds/AM or PM areas, or by highlighting an area and using the up and down arrows to select new values. The default is the current time.
6. Optionally, select an end date and time by selecting the Specify End Date check box. Specify the end date by typing the date in the month/day/year areas, or by clicking the down-arrow to open a calendar from which you can choose a date. Specify the time by typing values directly into the hour/minute/seconds/AM or PM areas, or by highlighting an area and using the up and down arrows to select new values.
7. Configure the scheduler to run on specific days of the week or month, on the last day of the month, or specify the beginning and ending times of the daily time span.
8. Close the pane.
9. Click the *Save* button to update the process flow.

Tip: If you wish to stop a query from executing regularly, you do not need to change the schedule. Instead, set Schedule Status to *Inactive* in the Schedule attribute pane. For details about setting flow properties, see *Setting Flow Properties* on page 144.

Procedure: How to Schedule a Process Flow to Run on Holidays

You can schedule a process flow to execute or not execute on holidays. The Holiday option is available when Recurring or Multi-Day are selected in the Schedule Type drop-down menu. For more information, see *Specifying a Holiday File* on page 413.

Note: You can only schedule the flow to run on holidays after you create a holiday file.

1. Right-click the *Start* object in the Process Flow tab and click *Properties*.

   The Properties pane opens.

2. Select *Active* from the Schedule Status drop-down menu.
3. Select *Recurring* or *Multi-Day* from the Schedule Type drop-down menu.
4. Select an option from the Run on Holidays drop-down menu. The following options are available:

   Skip

   Excludes a run on desired holidays.

   Only

   Only run on desired holidays.

   Run

   Only available if the schedule type is set to *Multi-Day*. Runs on all holidays.

5. Click the *Save* button to update the process flow.

To verify that the holiday schedule is set, in the navigation pane, expand the *Workspace* folder. Then expand the *Special Services and Listeners* folder. Right-click *SCHEDULER* and click *Scheduled Events*. The Scheduler/Scheduler Events report should list scheduled flows by the holiday set date.

Procedure: How to Schedule a Process Flow to Run When the Server Starts

You can schedule a process flow to run whenever the server is started. This option is useful that need to run all of the time, for example, when using Change Data Capture or File Listener capabilities.

1. Right-click the *Start* object in the Process Flow tab and click *Properties*.

   The Properties pane opens.

2. Select *Active* from the Schedule Status drop-down menu.
3. Select *Run when server starts* from the Schedule Type drop-down menu.
4. Click the *Save* button to update the process flow.

Procedure: How to Disable a Schedule Object

You can disable a the schedule of a flow from the Properties pane for a schedule.

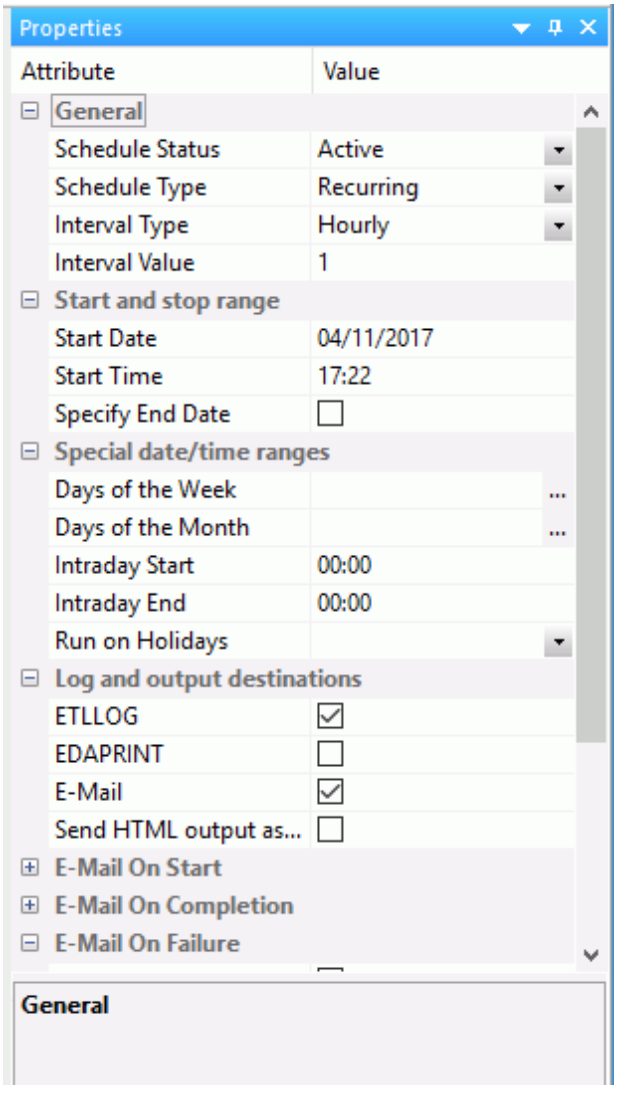
1. Right-click the *Start* object in the Process Flow tab and click *Properties*.

The Properties pane opens.

2. Select *Inactive* from the Schedule Status drop-down menu.
3. Close the pane.

Reference: Schedule Attribute Pane

To access the Schedule attribute pane shown in the following image, right-click the *Start* object and click *Properties*.



The Schedule attribute pane contains the following fields and options:

Schedule Status

Determines if there is a schedule and if so, whether the schedule is active or inactive.

Schedule Type

Determines whether a process flow will execute once, on a recurring basis, on several specific days (Multi-Day), or when the server starts.

Interval Type

Specifies the interval in which a process flow will be executed. For example, if you select Month as your Interval Type, and select 2 as your Interval Value, the flow will execute every two months. This option is only available if the schedule type is set to *Recurring*.

Interval Value

Specifies the frequency at which a process flow will be executed. For example, if you select Week as your Interval Type, and select 2 as your Interval Value, the flow will execute every two weeks. This option is only available if the schedule type is set to *Recurring*.

Start Date

Specifies the start date for the execution of the process flow.

Start Time

Specifies the start time for the execution of the process flow.

Specify Stop Time

Allows you to specify an end time for the execution of the process flow. When selected, it forces the job to end the execution by the time specified in Stop Time. This option is available if the Schedule Type is set to *Run Once* or *Multi-Day*. This option is also available if the Schedule Type is set to *Recurring* and Interval Type is set to Daily, Weekly, Monthly, or Yearly.

Specify End Date

Allows you to specify an end date and time for the execution of the process flow. This option is only available if the schedule type is set to *Recurring* or *Multi-Day*.

End Date

Specifies the end date for the execution of the process flow. This option is available if Specify End Date is selected.

End Time

Specifies the end time for the execution of the process flow. This option is available if Specify End Date is selected.

Days of the Week

Specifies the day of the week on which the process flow will be executed. This option is only available if the schedule type is set to *Recurring* or *Multi-Day*.

Days of the Month

Specifies the day of the month on which the process flow will be executed. This option is only available if the schedule type is set to *Recurring* or *Multi-Day*.

Intraday Start

Specifies the beginning of the daily time span of the process flow. This option is only available if the schedule type is set to *Recurring*.

Intraday End

Specifies the end of the daily time span of the process flow. This option is only available if the schedule type is set to *Recurring*.

Run on Holidays

Specifies the option to execute the process flow on desired holidays.

The following options are available:

- **Skip.** Excludes run on desired holidays.
- **Only.** Only run on desired holidays.

This option is only available if the schedule type is set to *Recurring* or *Multi-Day*.

Log and output Destinations

This option controls where the log information from the Maintenance Job should be sent when the Job finishes.

The following options are available:

- **ETLLOG.** The log is sent to the Data Migrator log.
- **EDAPRINT.** The log is sent to the Server log.
- **E-Mail.** The log is sent as an attachment to an email message if specified in Job Properties.

  Multiple options can be selected. The default values for this option are ETLLOG and E-Mail.

  Note: If the job is run from the DMC or Web Console, information will go to the console log regardless of this setting.

- **Send HTML output as inline message.** When selected, if any stored procedures called by the flow generate an output, the message is included as message text, instead of as an attachment.

  Note: This option for a flow schedule works in conjunction with the *Merge formatted output into log* option that is available for a Scheduler Special Service. The following considerations should be taken into account:

  - When the *Send HTML output as inline message* and *Merge formatted output into log* options are selected, the submitted flow emails the attached ETLLOG file and the HTML output is embedded into the body of the email.
  - When the *Send HTML output as inline message* and *Merge formatted output into log* options are not selected, the submitted flow emails the attached ETLLOG file and the HTML output is embedded into the attached ETLLOG file

Reference: Specifying a Holiday File

You can specify a list of dates that are designated as holidays in your company. These dates can be scheduled to be excluded, only included, or included to run with regular days in the process flow. To define a list of holidays, you must create a holiday file using a standard text editor and select the holiday file by issuing the SET command with the HDAY parameter.

1. In a text editor, create a list of dates designated as holidays using the following rules for creating a holiday file:
   - Dates must be in YYMD format.
   - Dates must be in ascending order.
   - Each date must be on its own line.
   - You may include an optional description of the holiday, separated from the date by a space.

   For example: 20160315 Ides of March
2. Save the file into the dm\etc directory:

   In Windows and UNIX the file must be HDAYxxxx.err.

   In z/OS, the file must be a member of ERRORS named HDAYxxxx, where xxxx is a string of text four characters long.
3. From the navigation pane, expand the *Workspace* folder. Expand the *Configuration Files* folder and right-click *Server Profile - edasprof.prf*. Click *Edit* and type:

   ```
   SET HDAY = xxxx
   ```

where:

xxxx
Is the part of the name of the holiday file after HDAY. This string must be four characters long. For example, if your file name is hdaymyh1.err, then the setting will be SET HDAY = myh1.

Note: Alternatively, instead of the server profile, the setting can be saved in the user profile for the scheduler as set in sched_scan_id.

4. Save the new Server Profile.
5. Test that holiday is correctly set up by creating a sample hdayfile synonym.

```
FILENAME=HDAYFILE, SUFFIX=FIX     ,
 DATASET=C:\ibi\srv82\dm\etc\hdaymyh1.err, $
  SEGMENT=HDAYFILE, SEGTYPE=S0, $
    FIELDNAME=HDAY_DATE, ALIAS=hddate, USAGE=YYMD, $
    FIELDNAME=HDAY_DESC, ALIAS=hddesc, USAGE=A66, ACTUAL=A68, $
```

Note: DATASET should be set to your appropriate directory path.

6. Run Sample Data for hdayfile synonym. An example of the sample data is shown in the following image.

	HDAY_DATE	HDAY_DESC
1	2019/01/01	New Year's Day
2	2019/01/21	Martin Luther King, Jr. Day
3	2019/02/18	Presidents' Day
4	2019/05/27	Memorial Day
5	2019/07/04	Independence Day
6	2019/09/02	Labor Day
7	2019/11/11	Veterans Day
8	2019/11/28	Thanksgiving Day
9	2019/11/29	Day After Thanksgiving
10	2019/12/25	Christmas Day

Chapter 7

Tutorial: Creating Data and Process Flows

Your company is trying to empower both its sales force and its line managers with timely and focused information to support needed decisions:

- Sales agents need information that will enable them to report quickly on the most and least popular products, and to track regional trends.
- Line managers need to assess strengths and weaknesses in their sales approaches using comparative analysis of competing stores to determine, for example, if the other stores are outselling them on one type of product. If so, they may need to initiate a sales promotion or provide more sales staff training.

Both groups need current and customized sales data to create self-service reports and perform analyses. It is your job to provide that data.

To accomplish these goals, you will create a flow that moves data out of an operational data store and that builds a denormalized table that can be used to run sales and analytic reports.

Note: Refer to the dmdemo sample data flow in ibisamp for the complete example. However, note that dmdemo has only one data target, while the flow you build will contain two.

In this chapter:

- What You Are Going To Do
- Startup and Setup Tasks
- Exercise 1: Creating the Data Flow
- Selecting and Joining Data Sources
- Selecting and Managing Columns
- Adding Data Targets
- Submitting the Data Flow
- Data Flow Review
- Exercise 2: Creating the Process Flow
- Starting the Process Flow
- Adding a Data Flow Object to the Process Flow
- Branching in the Process Flow
- Submitting the Process Flow

What You Are Going To Do

The Data Migrator tutorial guides you through the creation of two flows:

- **In Exercise 1, you create and save a Data Flow** that extracts data from two data sources, and then transforms and loads the extracted data into two data targets. You will save the data flow and use it again in Exercise 2.
- **In Exercise 2, you create and save a Process Flow** that uses the data flow from exercise 1 as a component, along with two stored procedures that log the success or failure of this processing. The flow manages this sequence of events.

Startup and Setup Tasks

This tutorial assumes that your Data Migrator Client and Reporting Server software have been installed and configured.

You will need to complete the following startup and setup tasks before you begin to design your data flow:

- **Start your Server.**
- **Start the Data Management Console.**
- **Add a server node** that points to the Server (optional). You can use this node instead of the one that was configured during the installation process.
- **Connect to the server.**
- **Create the sample data and metadata** you will be using as data sources and data targets in the tutorial.
- **Set up the sample application directory** you will use in the tutorial. You create the new application and configure its path from the Web Console. You will be storing the flows and associated files you create in this application.

Start Your TIBCO WebFOCUS Reporting Server and Data Management Console

Using whatever method is appropriate for your operating system:

1. Start your Server.
2. Start the DMC.
3. If you are not going to configure an additional server node, right-click the default LOOPBACK node in the navigation pane and click *Connect*.

Configure a TIBCO WebFOCUS Reporting Server Node (Optional)

For the tutorial, we will create an additional node and name it DEMO. We will store and work with flows and related files in this location.

Note: Creating an additional node is only required if the server is remote.

Since many users prefer, or need, to establish additional server nodes that point to the Reporting Server they will be using in the course of their work, we have added this setup task.

However, you may wish to use the Server node that you configured during the Data Migrator Client installation. The node is simply a name that, by way of its properties, points to the Reporting Server.

If you wish to use the node name you set up during the Data Migrator Client installation skip to *Create Sample Data and Metadata and Create an Application Directory* on page 419. Keep in mind that when an instruction in the tutorial refers to DEMO, you must apply the instruction to the name of the server node you are using.

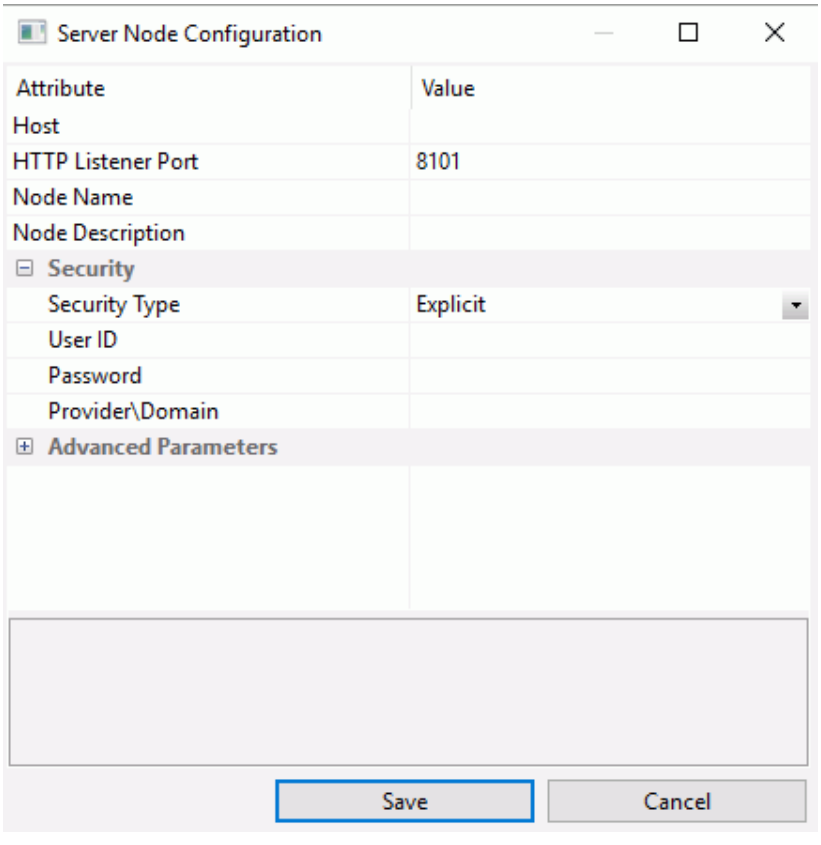
1. If the DMC is not already open, select*Information Builders**WebFOCUS 82 Client or Server*, and then click *Data Management Console* from the Windows Programs menu. The DMC opens, with the server node you configured during installation listed under Servers in the navigation pane at the left.

   The black circle that appears on the server icon indicates that the DMC is not currently connected to the Server. You will connect shortly.

   Notice that the Console Log at the bottom of the screen has tracked your startup steps. It will continue to track each step in your work process.

2. For the tutorial, we will name the new node DEMO. To make the DEMO server available, you will define the server node.

Right-click *Servers* and click *Add Server Node*. The Server Node Configuration dialog box opens, as shown in the following image.



3. Enter the following settings:

   - ❏ In the Host field, enter the machine name or IP address where the Reporting Server is running.
   - ❏ In the HTTP Listener Port field, enter the Port number of the Server. If you used the defaults when you installed your Reporting Server, the port number is 8121.
   - ❏ In the Node Name field, enter *DEMO* as the name for the Server node.
   - ❏ Optionally, enter a description of the node in the Node Description field.
   - ❏ Leave the Security field set to *Explicit*.
   - ❏ Enter a valid user ID and password, and if required, a domain or provider name.

4. Click *Save*.

   The DEMO node is now added to the Servers list.

5. Right-click the DEMO server node in the navigation pane, and click *Connect*.

Create Sample Data and Metadata and Create an Application Directory

Your first task is to create the sample data and metadata you will be using in the tutorial.

To create the sample data and metadata from the DMC, see *How to Create Sample Procedures and Data* on page 59.

After you have created your sample data, set up a separate application directory called *salesdemo* in the Application Directory tree. This application directory will contain the flows and other files you will be creating.

1. Connect to the DEMO server using a user ID with server administrator privileges.
2. Under the DEMO server, right-click *Application Directories*, point to *New*, and then click *Application Directory*.

   The Create New Application dialog box opens.

3. In the Application Name box, enter *salesdemo* as the new application directory name and click *OK*.

   The salesdemo application directory is now listed in the approot tree in the navigation pane. Notice that ibisamp, the directory in which sample data is installed by default, is already in the Application Path box, which means it is in the application path of the server.

   If the directory is not listed, right-click *Application Directories* and click *Application Path*. Check that the check box in the In Path column is selected for salesdemo. Click *Save* to save your changes.

   Tip: In addition to salesdemo and ibisamp, the Application Directory tree contains a directory called baseapp. This is a default directory that is added automatically to your Application Path during installation. There is no need to add baseapp explicitly to your Application Path. The tree also contains a folder called foccache (Temporary) which contains temporary files for the duration of your session.

   The salesdemo folder you created in the Web Console now appears in the Application Directories list.

Next step: Your setup is complete. The next step is to create a data flow.

Exercise 1: Creating the Data Flow

Your setup is done, and you are ready to start building the Data Flow.

Right-click the *salesdemo* application directory, point to *New*, and then click *Flow*.

A workspace opens in the right pane. The Data Flow tab is active and an SQL Select Columns object is displayed in the center of the pane.

Next step: You will start by selecting the data sources you will be using in this data flow.

Although you will be creating your flows in the salesdemo application directory, you will be using sample files from the ibisamp application directory (since this is where they are stored when you create the sample procedures).

Selecting and Joining Data Sources

Your first task is to specify the operational data sources from which you will be extracting and moving data. You will be adding two data sources to the data flow and joining them. In one of the data sources, you will create a virtual column, called a source transformation in Data Migrator terminology. This transformation will exist only in the data flow object where you build it.

1. In the DMC navigation pane, expand the *ibisamp* application directory.
2. On the Home tab, in the Filter group, click *Synonyms*.
3. Scroll down to the files that begin with the letters *dm*. All of these files were created in the data adapter of your choice when you chose *Create Sample Procedures and Data* (see *Create Sample Data and Metadata and Create an Application Directory* on page 419).

   You will be using three synonyms from this group. Synonyms are used to describe both data sources and data targets. The first two are used for data sources and the last one for a data target:

   dminv

   dmord

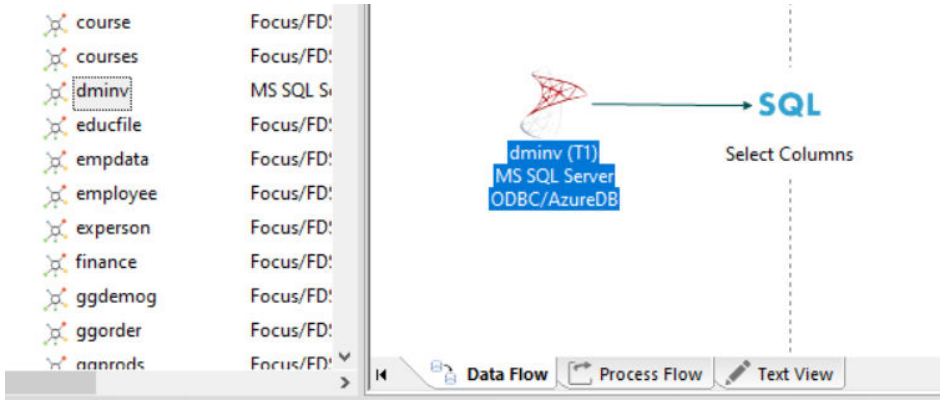
   dmrpts

Tip: At many sites, data sources and data targets are on different servers and platforms. For this tutorial, we have chosen the simplest, most universal configuration. For information about different configuration options, see *Remote TIBCO WebFOCUS Reporting Server Setup At a Glance* on page 30 and *Planning a TIBCO Data Migrator Application* on page 87.

Add Your First Data Source

You will start by selecting the synonym for the inventory table called *dminv* from the expanded ibisamp directory.

1. Drag *dminv* from the ibisamp application directory into the Data Flow workspace, to the left of the SQL object, as shown in the following image.



A connecting arrow is automatically added between these objects.

Tip: If you drag a synonym object to the left of the SQL object, it is understood to be a data source. If you drag it to the right of the SQL object, it is understood to be a data target.

2. Let us take a quick look at the data in the table you selected. By default, you can retrieve 50 rows from the server to verify that this is the correct table and that the data is available.

Right-click the *dminv* object in the workspace, point to *Operations*, and then click *Sample Data*. The sample data for the table opens, as shown in the following image.

	Product Number	Product Name	Quantity in Stock	Sale Price	Our Cost	Product Category
1	1004	2 Hd VCR LCD Menu	43,068	179.00	129.00	VCRs
2	1006	Combo Player - 4 Hd VCR + DVD	13,527	399.00	289.00	DVD
3	1008	DVD Upgrade Unit for Cent. VCR	199	199.00	139.00	DVD
4	1010	750SL Digital Camcorder 300 X	10,758	999.00	750.00	Camcorders
5	1012	650DL Digital Camcorder 150 X	2,972	899.00	710.00	Camcorders
6	1014	340SX Digital Camera 65K P	990	249.00	199.00	Cameras
7	1016	330DX Digital Camera 1024K P	12,707	279.00	199.00	Cameras
8	1018	250 8MM Camcorder 40 X	60,073	399.00	320.00	Camcorders
9	1020	150 8MM Camcorder 20 X	5,961	319.00	240.00	Camcorders
10	1022	120 VHS-C Camcorder 40 X	2,300	399.00	259.00	Camcorders
11	1024	110 VHS-C Camcorder 20 X	4,000	349.00	249.00	Camcorders
12	1026	AR3 35MM Camera 10 X	12,444	129.00	95.00	Cameras
13	1028	AR2 35MM Camera 8 X	11,499	109.00	79.00	Cameras
14	1030	QX Portable CD Player	22,000	169.00	99.00	CD Players
15	1032	R5 Micro Digital Tape Recorder	1,990	89.00	69.00	Digital Tape Re

The information has been retrieved, so you know the data source is available. Close the sample window by clicking the *X* in the tab above it.

Add and Join a Second Data Source to the First

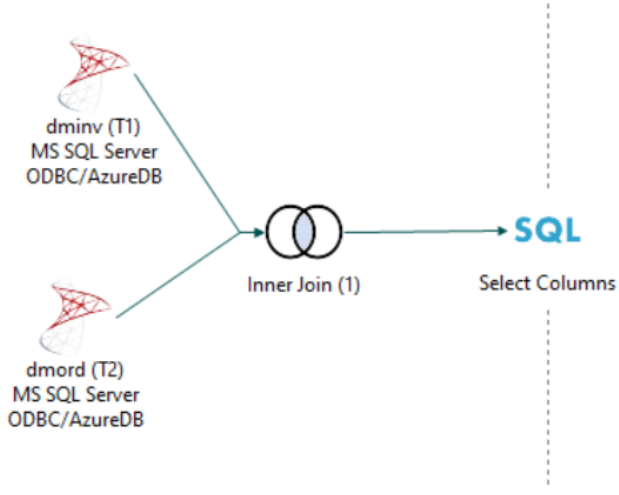
To add another data source to the flow:

1. Right-click in the workspace to the left of the SQL object and click *Add Source*. The Select Synonym dialog box opens.

   If the ibisamp directory is not selected in the Look in: section, click it in the tree.

2. Scroll down and select the order table called *dmord*. Then click *Select*. (You can also drag dmord from the navigation pane, as you did earlier with dminv.)

The dmord source object is now connected to the dminv source object by way of a join object, which is linked to the SQL object, as shown in the following image.

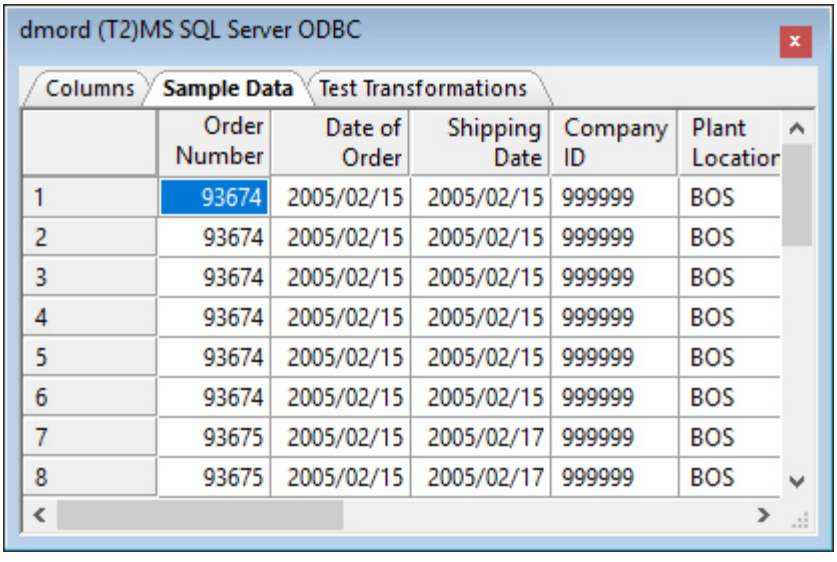


Tip: The connecting arrows between these objects are added because the option *Automatically add join conditions* is selected by default. If these connections are not appearing on your screen, go to the Home tab and, in the Tools group, choose *Options*. Click *Data Flow Designer* under Data Flow in the navigation tree, and then click *Automatically add join conditions*.

3. We will verify the data in the orders table using a different method.

   Right-click the *dmord* source object and click *Expand*. The Columns tab shows the columns in the data source.

Click the *Sample Data* tab to verify that data is being retrieved from the orders table, as shown in the following image.



	Order Number	Date of Order	Shipping Date	Company ID	Plant Location
1	93674	2005/02/15	2005/02/15	999999	BOS
2	93674	2005/02/15	2005/02/15	999999	BOS
3	93674	2005/02/15	2005/02/15	999999	BOS
4	93674	2005/02/15	2005/02/15	999999	BOS
5	93674	2005/02/15	2005/02/15	999999	BOS
6	93674	2005/02/15	2005/02/15	999999	BOS
7	93675	2005/02/15	2005/02/17	999999	BOS
8	93675	2005/02/15	2005/02/17	999999	BOS

The first 50 rows are retrieved.

Tip: You can change the number of rows to retrieve from the Tools group. On the Home tab, in the Tools group, click *Options*. From the Options dialog box, click the *Run Options* link, and then change the number in *Maximum number of rows for test reports*.

4. Double-click the title bar or click the *X* in the upper-right corner to close the window and return to the object view.

   Note: You can also double-click a source object and click the *Sample Data* button in the Source Transformations window. You can place sample data in the workspace by right-clicking a source object, pointing to *Operations*, and clicking *Sample Data*.

Create A Virtual Column in the Second Data Source

Shortly, we will take a closer look at the join you created. Before that, you will create a virtual column in the dmord source object (this is also called a source transformation). Data transformations in a source object are performed when the records are read, before any filtering or aggregation occurs.

Planning ahead a bit, you know that the time frame for the data target you will be creating is the year and month. To facilitate the mapping of source to target data, you decide to create a virtual column called YEARMONTH in the dmord data source, to be derived using the expression YEAR x 100 + MONTH. This column will represent both the year and the month.

1. Double-click the *dmord* source object. (You could also right-click the *dmord* source object and click *Source Transformations*.) The Source Transformations dialog box displays all of the columns in the data source.
2. To define the virtual column, click the *Insert transforms* button.

   The Transformation Calculator opens.

   a. In the Name box, type *YEARMONTH*.

   b. In the Format box, type *I6* (I as in integer).

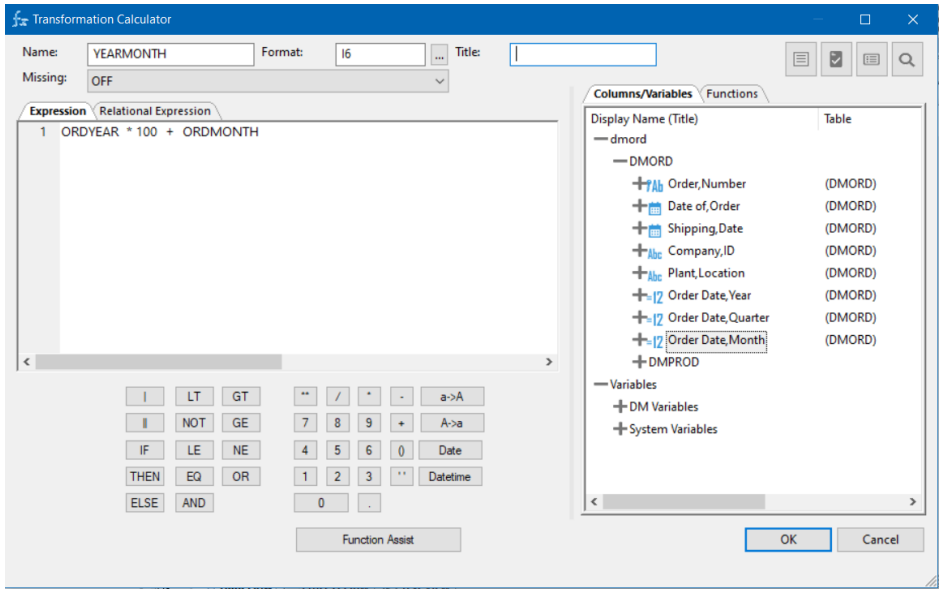
   c. In the Columns/Variables tab, double-click *Order Date,Year*, or *ORDYEAR* if viewing by Name.

   d. On the calculator keypad, click the multiplication sign (*).

   e. In the Expression box, type *100* +.

   f. Double-click *Order Data,Month*, or *ORDMONTH*.

   The calculator should look like the image below:

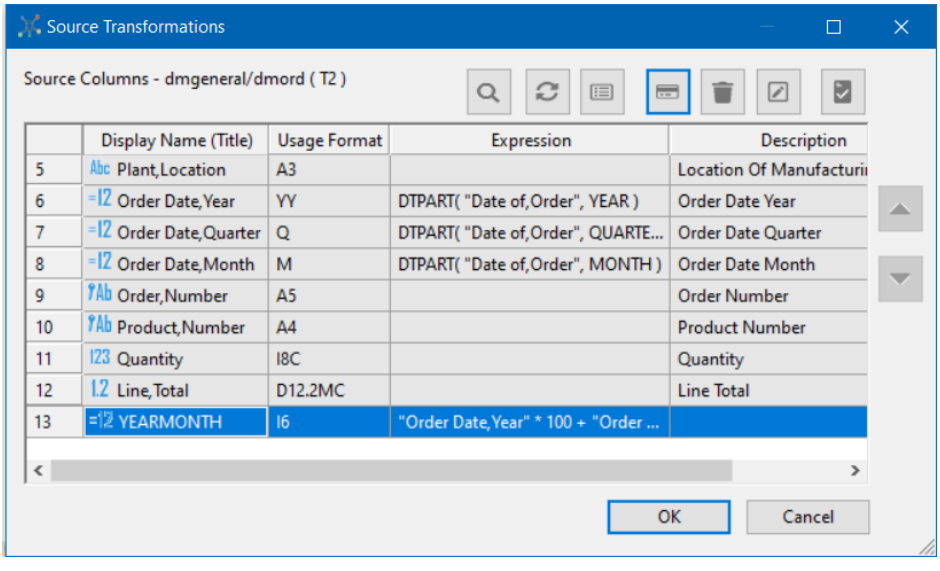


Tip: Before you close the calculator, notice that the two columns you used in the expression are themselves virtual fields (or transformations) that were defined directly in the synonym.

Virtual columns are identified by the symbol = (with a format identifier). Adding a virtual column in a synonym is a useful strategy when you expect to use the same synonym with more than one flow. For details about adding virtual columns in a synonym, see *Adding Virtual Columns (DEFINE) in a Synonym* on page 527.

3. Click *OK* to close the calculator and return to the Source Transformations dialog box.

   Notice that YEARMONTH now appears in the Source Columns list, as shown in the following image.



4. To verify that the transformation is working properly, click the *Test Transforms* button to view data.
5. You will now be able to use YEARMONTH as source data.

   Close the Test Transformation dialog box, and then click *OK* to close the Source Transformations dialog box.

Assign Join Properties

Now let us consider the join object, which was added automatically when you selected the second data source. You will need to specify properties for the join.

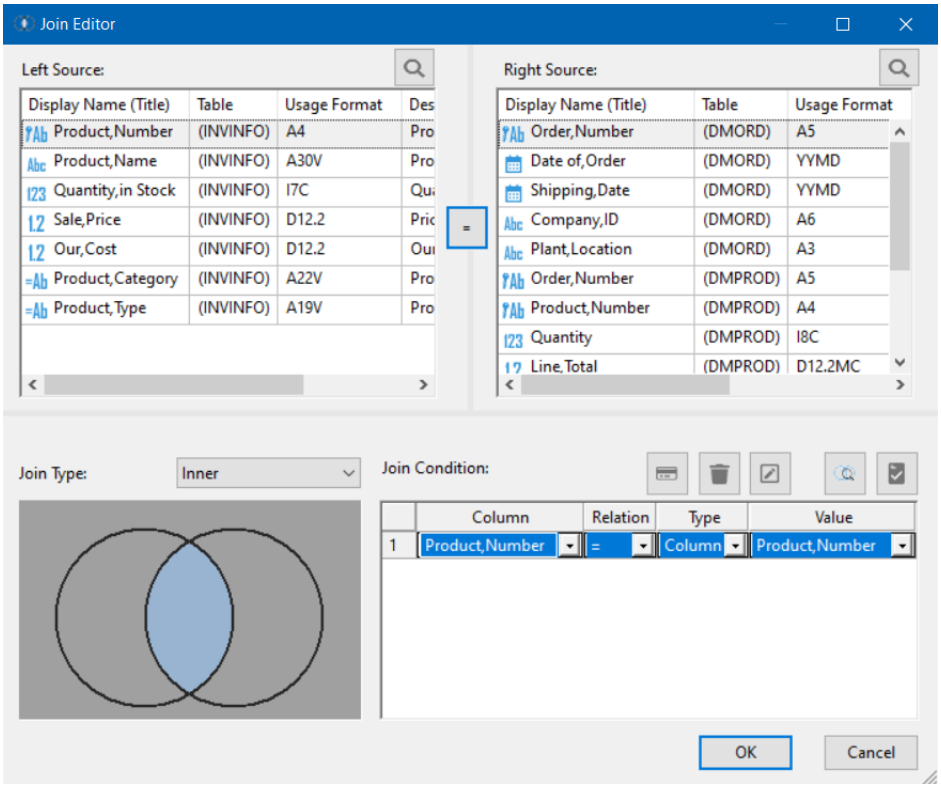
By default, an inner join is created. An inner join extracts those rows that appear in both tables. You will base the join on an equality condition between two fields, one in each data source. The use of an equality condition is also called an equi-join.

Tip: Although not illustrated in this tutorial, Data Migrator supports multiple joins, joins based on conditions other than equalities, and joins that are modified by calculations, such as substrings or concatenations. A Join Calculator is available to assist you.

1. Right-click the join object and click *Join Editor*. The Join Editor dialog box opens.
2. The join must be based on columns in each of the joined data sources. Notice that *Product,Number (PROD_NUM)* is in both Left and Right Source Columns lists. The join of T1.PROD_NUM and T2.PROD_NUM appears in the Expression field of the Join Conditions list. For the tutorial, the default join on PROD_NUM is sufficient.

   Tip: Once again, the automatic join condition is in effect because the option *Automatically add Join conditions* is selected by default. If this is not the case, on the Home tab, in the Tools group, click *Options*. In the Options dialog box, click the *Data Flow Designer* link and select *Automatically add Join conditions*.

The inner join relationship is reflected in the Join Condition section and is represented graphically by the overlapping area in the Join Type diagram, as shown in the following image.



Tip: If you want to change the type to a right or left outer join, you can simply click the left or right circle. Try that now if you like, and then return to the inner join position. For details about join options, see *Joining Data Sources* on page 204.

3. Click *OK* to close the Join Editor.

Next step: You have now set up your source data. You are ready to select the columns from the source data to load into the data targets, which you will create shortly. To select and structure the columns, you will be using the SQL object mentioned earlier.

Selecting and Managing Columns

You are now ready to select the columns of data you want to load into the data targets. You can perform a variety of operations on the selected columns.

Right-click the SQL object and notice the options on the menu.

- **Expand/Collapse.** Opens or closes an information window, as you saw earlier.
- **SQL Statement.** Displays the SQL code. Right now it simply reflects the join. It will be more interesting in a little while.
- **WHERE Filter.** Provides a calculator in which you can create an expression that limits record selection. For example, you might only want to retrieve records for a certain year.
- **HAVING Filter.** Provides a calculator in which you can create an expression that limits retrieval based on aggregated values after a GROUP BY. This option only appears when the Column Selection includes a GROUP BY.
- **Sort.** Provides a dialog box in which you can control the order in which data is retrieved.
- **Column Selection...** Opens a window in which you select the columns you want to include in your data target, and specify a variety of data retrieval requirements, as you will do in the following steps.
- **Add.** Allows you to add additional sources, selects, joins, and union objects.
- **Properties.** Opens a property panel that shows statistics for the selected statement.
- **Error Details.** This option only appears if there is an error in the columns or filters. When selected, a dialog box opens to explain the error.

Select and Create Columns

1. Right-click the *SQL* column object and click *Column Selection*.

   The Column Selection dialog box opens. You can identify column types at a glance:

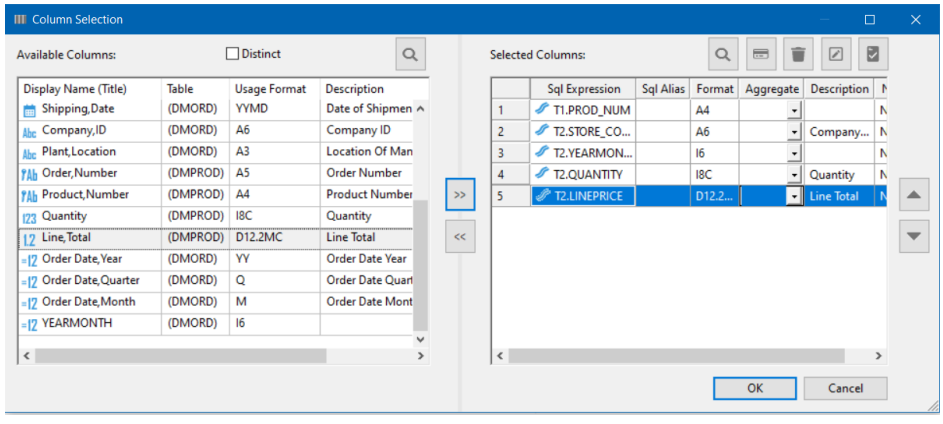
Symbol	Description
	Key column **Note:** The icon will vary based on the format of the column.
	Regular column **Note:** The icon will vary based on the format of the column.

Symbol	Description
	Index column
	Virtual column The symbol for columns in the synonym is gray, while the background for a source transformation (virtual column in the flow) is white. **Note:** The icon will vary based on the format of the column.
	Source Transformation This is only available in the Source Transformations dialog box. **Note:** The icon will vary based on the format of the field.

2. You want the sales force to be able to track and evaluate product and store information, so double-click the columns *Product,Number (PROD_NUM)* from INVINFO, and *Company,ID (STORE_CODE)* from DMORD in the Available Columns list. This moves them into the Selected Columns list.
3. Next, you want to specify a time period. In this case, you want to select YEAR and MONTH, to provide the sales force with the ability to analyze sales on a monthly basis.

   Double-click the column *YEARMONTH* in the Available Columns list to move it into the Selected Columns list. (This is the column you created earlier as a Source Transformation.)

4. Now, add the columns you will use to measure sales. Double-click *QUANTITY* and *Line,Total (LINEPRICE)* in the Available Columns list to add them to the Selected Columns list. The Column Selection window should now look like the image below:



5. Next, you want to calculate the line item cost of goods. This transformation can be done using an SQL calculation to create the column.

   To open the SQL Calculator, click the *Insert Columns* button above the Selected Columns list.

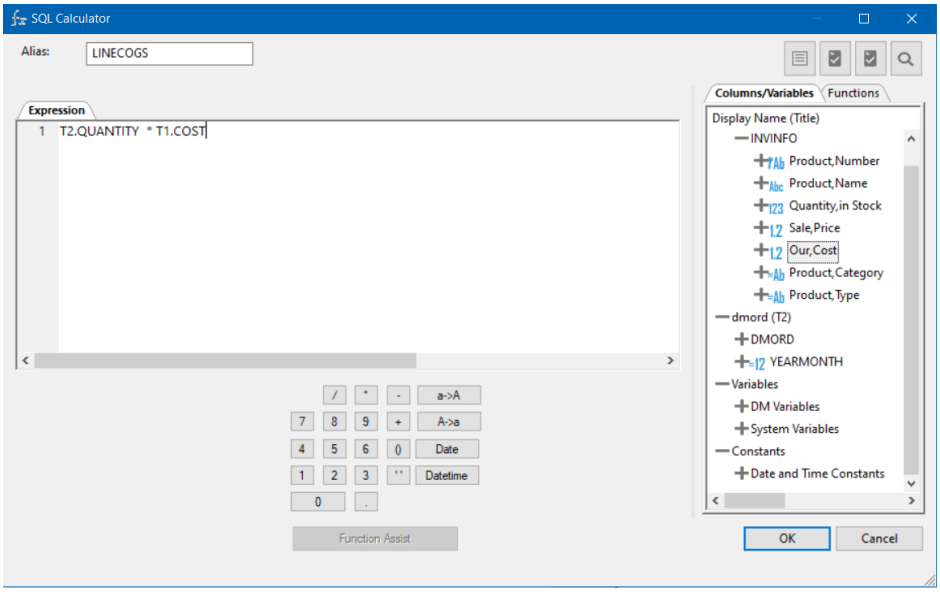
   a. In the Alias box, enter *LINECOGS*.

   b. Ensure that the Columns/Variables tab is selected, and then double-click *QUANTITY* in the tree. The column appears in the Expression box.

   c. Click the multiplication sign (*) on the calculator keypad.

   d. Double-click *Our,Cost (COST)* in the tree to complete the expression.

The SQL Calculator should look like the image below:



6. Click *OK* to close the SQL Calculator and return to the Column Selection dialog box.

   Notice that the expression has been added at the bottom of the Selected Columns list. The expression is in the first column and the alias you assigned is in the second column.

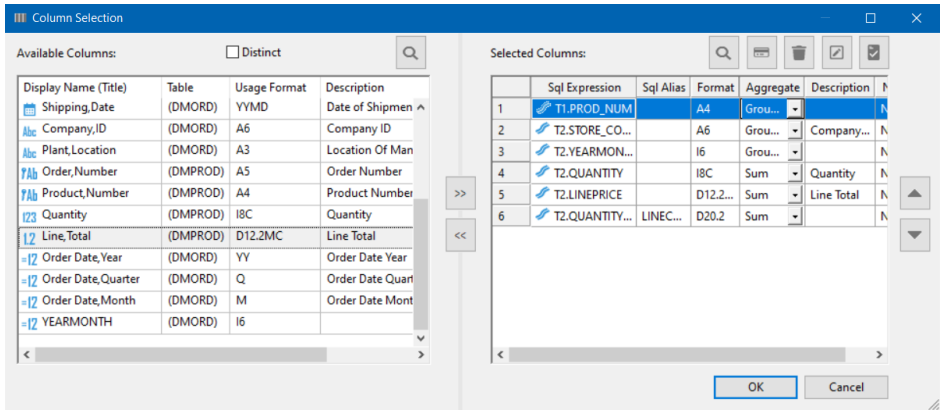
7. Since you want to generate month level detail, you must aggregate on the key columns, and then on the year/month column. This is done by adding the Group attribute to each column.

   Under Selected Columns, click *T1.PROD_NUM*, then select *Group By* from the drop-down menu in the Aggregate column.

   Repeat this step for *T2.STORE_CODE* and for *T2.YEARMONTH*.

8. Finally, let us sum the three measures: *T2.QUANTITY*, *T2.LINEPRICE*, and *LINECOGS*. You can multi-select them.

Under Selected Columns, press the Ctrl key and click T2.QUANTITY, T2.LINEPRICE, and LINECOGS. Choose *Sum* from the drop-down menu in the Aggregate column. Sum is applied to all three columns, as shown in the following image.



9. Click *OK* to close the Column Selection window and return to the object view in the Data Flow tab.

Filter the Data

You have identified the columns you want to extract from the source data, but the data in the source goes back several years, and only the last few years are really relevant to the salespeople who need to make product decisions for the coming year. Therefore, you are going to define a selection criterion to limit the retrieval to the appropriate years.

1. Right-click the *SQL* object again, and this time click *WHERE Filter*. The WHERE Filter Calculator opens to assist you in constructing the expression. Notice that this calculator is suitable for creating a wide range of selection criteria.

   As you can see, the columns here are represented by the same icons used in the Selected Columns dialog box. Notice that YEARMONTH is a virtual column, rather than a real database column. You can filter on real or virtual columns.

2. To limit record retrieval to the time period beginning in January, 2006:

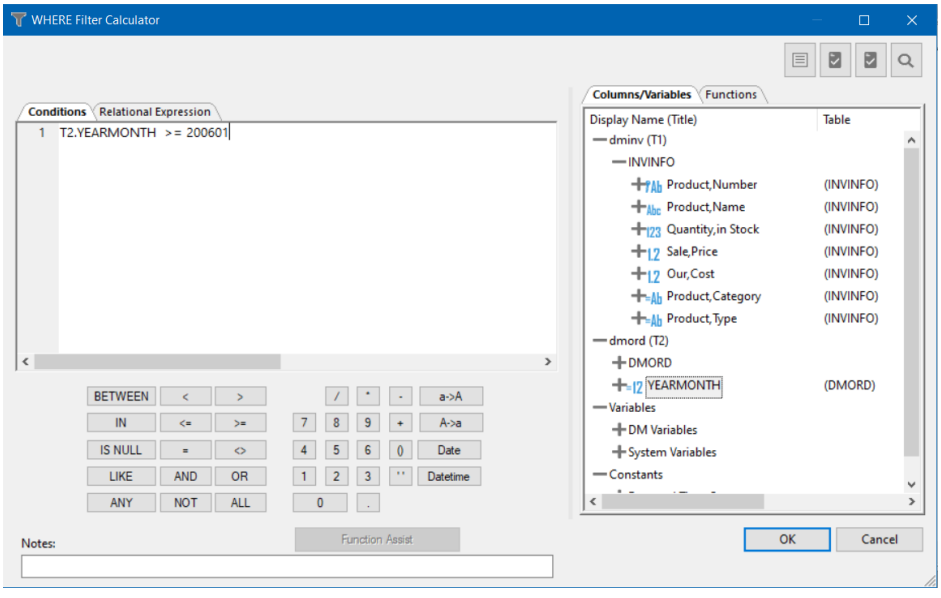
   a. Click the *Conditions* tab, if not already selected.

   b. Double-click *YEARMONTH* in the Column list. YEARMONTH is displayed in the Expression box.

   c. Click >= (greater than or equal to) in the calculator pad below the Expression window.

      Click the numbers *2 0 0 6 0 1* on the pad, or type *200601* directly into the Expression box to specify the year and month.

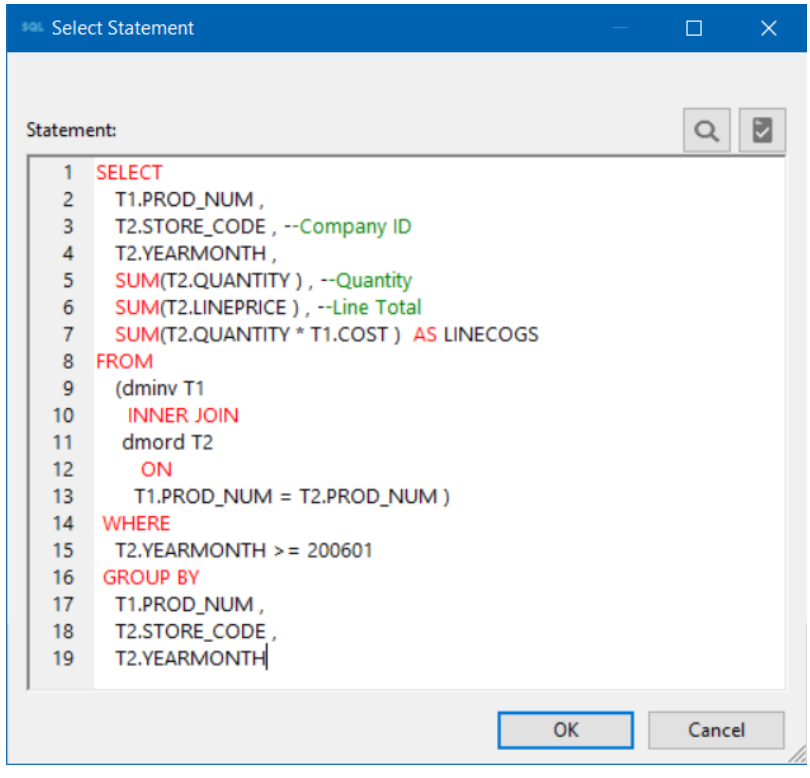
The expression in the WHERE Filter Calculator should look like the following image:



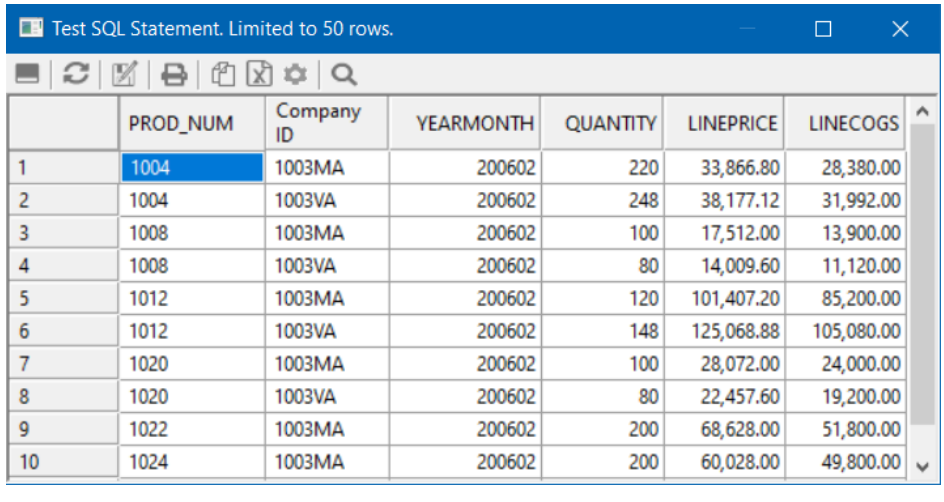
Tip: Though it is not required for this tutorial, you can build an SQL calculation using any ANSI SQL function. Simply click the *Functions* tab to see the available functions and arguments.

3. For now, click *OK* to complete the filtering expression.

4. Right-click the *SQL* object and click *SQL Statement*. The window should look like the following image:



5. Click the *Test SQL Statement* button to see the results, as shown in the following image.



	PROD_NUM	Company ID	YEARMONTH	QUANTITY	LINEPRICE	LINECOGS
1	1004	1003MA	200602	220	33,866.80	28,380.00
2	1004	1003VA	200602	248	38,177.12	31,992.00
3	1008	1003MA	200602	100	17,512.00	13,900.00
4	1008	1003VA	200602	80	14,009.60	11,120.00
5	1012	1003MA	200602	120	101,407.20	85,200.00
6	1012	1003VA	200602	148	125,068.88	105,080.00
7	1020	1003MA	200602	100	28,072.00	24,000.00
8	1020	1003VA	200602	80	22,457.60	19,200.00
9	1022	1003MA	200602	200	68,628.00	51,800.00
10	1024	1003MA	200602	200	60,028.00	49,800.00

Note: The number of records retrieved will depend on the Run Options set from *Tools* and *Options*, and the database you choose for your sample data.

6. Click the red X to close the Test SQL Statement window, and click *OK* to close the Select Statement window.

Next Step: You are ready to create the data targets into which your source data will be moved, based on the mapping and rules you defined in the SQL columns object.

Adding Data Targets

You are going to add two data targets and specify the options you want to use when loading data into them.

- The first data target is a pre-existing table for sales reports that we designed just for this purpose. You are going to add data from last year to it.
- The second is a target table that you will create to provide additional analytic data that will be useful to the line managers.

The newly created target will be updated by other flows.

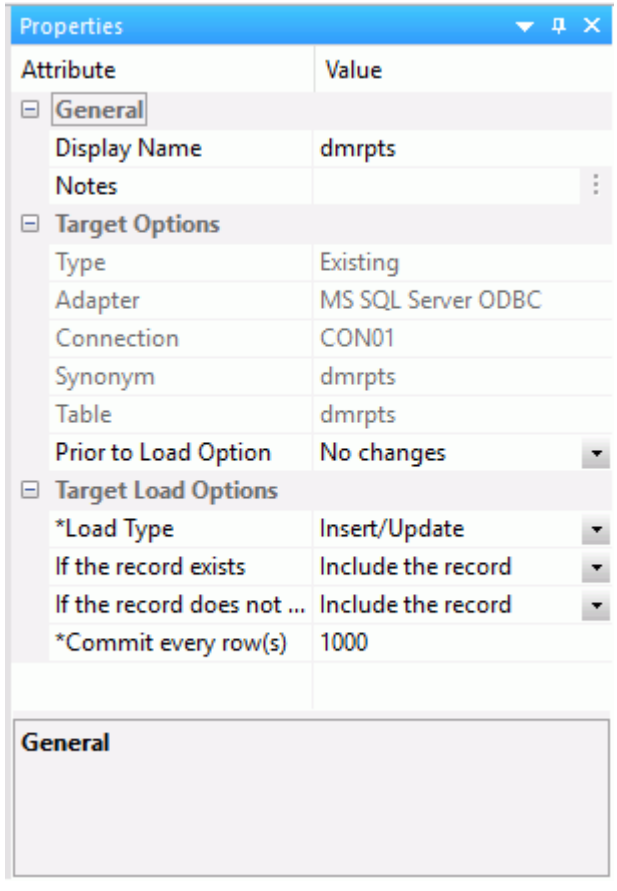
Note: Prior to adding the target, right-click the flow canvas and select *Flow Properties*. *Optimize Load* is selected by default. However, it does not support some of the target transformation options used in the tutorial. In the *Execution* section, deselect *Optimize Load* and close the Properties panel.

Map Columns in an Existing Data Target

To specify the first data target:

1. From the navigation pane, drag the synonym *dmrpts* from the ibisamp directory into the workspace, to the right of the SQL object. (The position to the right of the SQL object makes it a data target.)
2. Once you have added the data target to the data flow, you can specify how incoming data should be handled during the loading process.

Right-click the *dmrpts* target object and click *Properties*. For this exercise, you will accept the default Load Type, Insert/Update, as shown in the following image.



Tip: If you intend to run a flow more than once, it is advisable to clear the target before starting the run. This technique generally enhances performance in subsequent runs. In the *Prior to Load Option* in the Target Properties window, select the clearing option that suits the adapter type being used for your data targets. For example, for a relational table, the option is *Truncate table*. For a FOCUS/FDS or XFOCUS table, the option is *Drop table*.

3. Click *X* to close the Properties window.

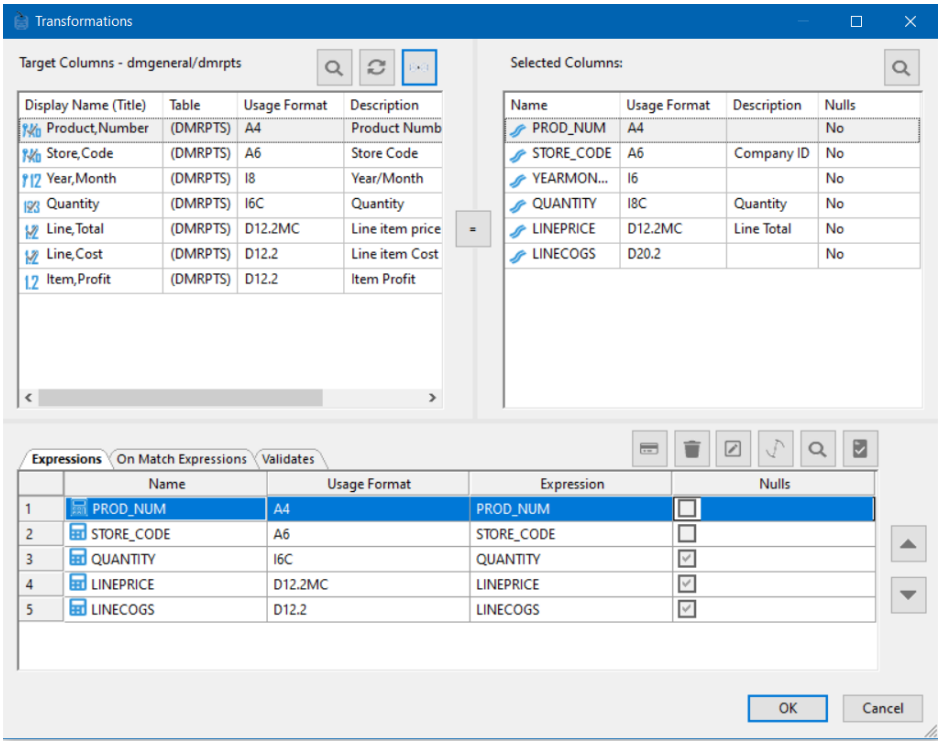
   The next step is to map the data source columns you are extracting to the data target columns.

4. Right-click *dmrpts* and this time, click *Target Transformations*. The Transformations dialog box opens.

Note: You may need to turn off Optimize Load by right-clicking in the flow workspace and deselecting that option in order to see all of the tabs on the Target Transformations dialog box.

5. Click the Automap button.

   The five columns with identical names and data types are mapped and moved to the Expression tab. Note that the mapped columns are also checked in the Target Columns list, as shown in the following image.



6. Two target column names are deselected, so you need to create mappings or transformations for them. In the Target Columns list, YRMTH contains year and month data. You will map it to the YEARMONTH column in the Selected Columns list.

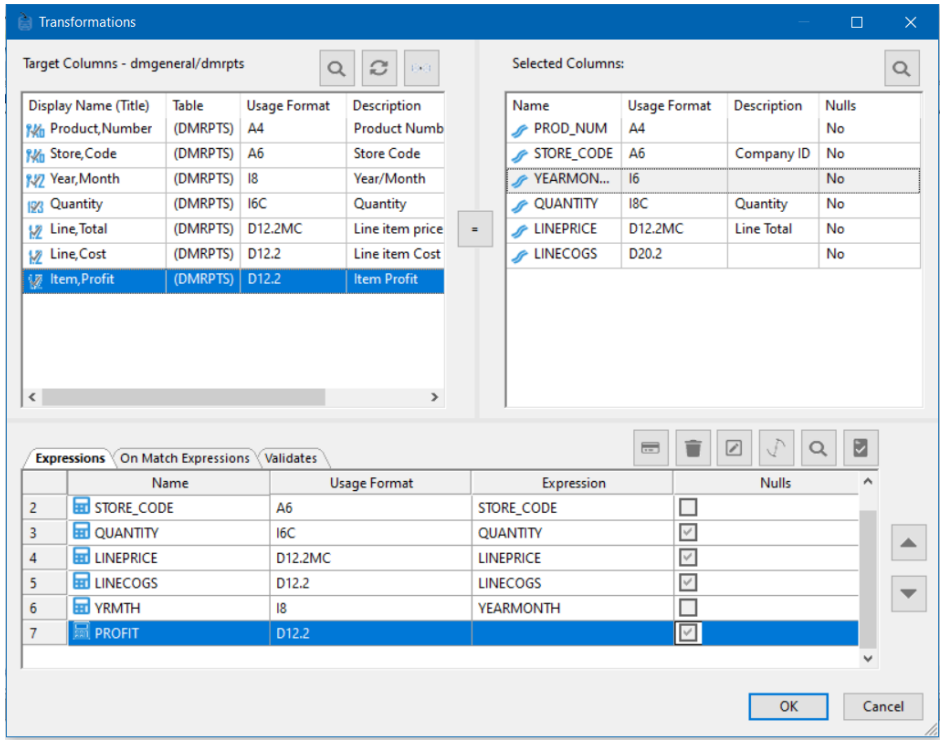
   a. Under Target Columns, click *Year,Month (YRMTH)*.

   b. Under Selected Columns, click *YEARMONTH*.

   c. Click the equal sign (=) to move the mapping into the grid in the Expressions tab.

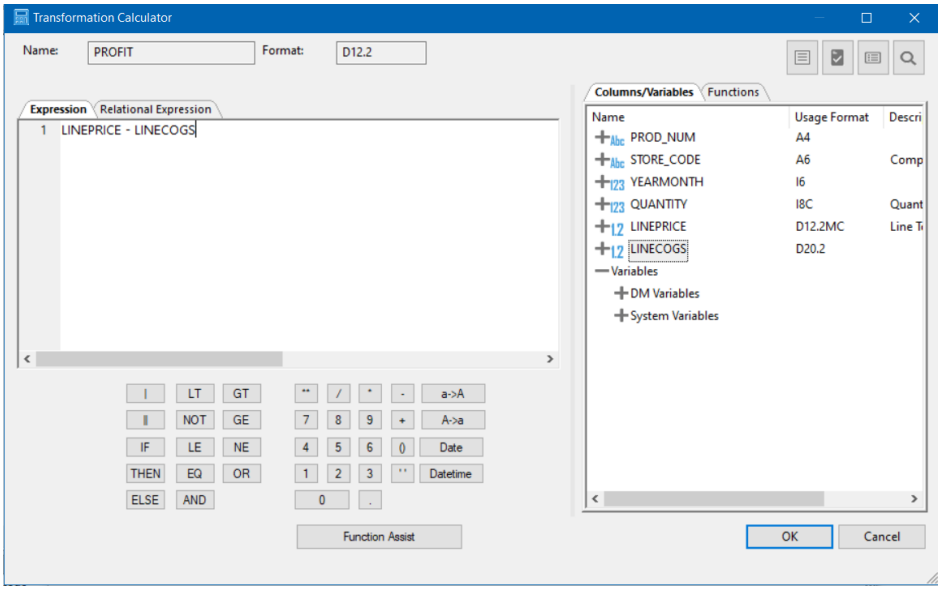
7. For the PROFIT column, you will build a transformation based on the definition of PROFIT as the difference between LINEPRICE and LINECOGS. As you can see, the data target is another place where we can apply transformations.

Under Target Columns, double-click *Item,Profit (PROFIT)* to move it to the Target Columns list in the Expressions tab, as shown in the following image.



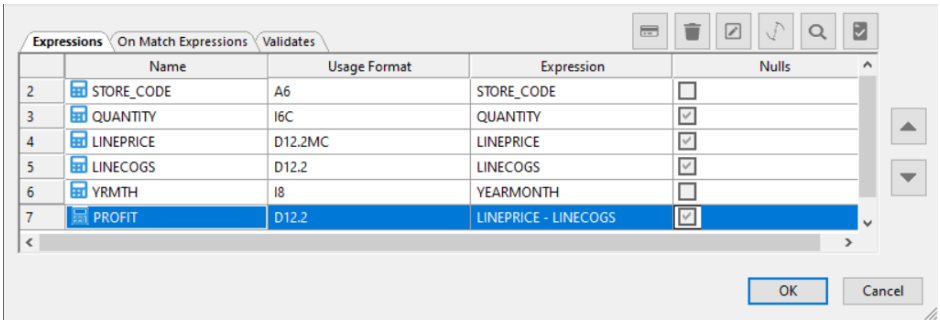
8. To open the Transformations Calculator, double-click *PROFIT* in the Expressions tab. To create the expression in the Expression box of the calculator:
   a. In the tree, double-click *LINEPRICE*.
   b. On the calculator pad, click the minus (-) sign.
   c. In the Source Columns list, double-click *LINECOGS*.

The Transformation Calculator should look like the following image:



9. Click *OK* to complete the expression and close the calculator.

   The Expressions tab of the Transformations dialog box now looks like the following image:



10. Scroll to the right to see the last column. It may be useful to validate the incoming data. To do that, load only those records with a quantity greater than or equal to ten. Records that do not meet this validation criterion can be logged to a file for additional processing or for review at a later date.

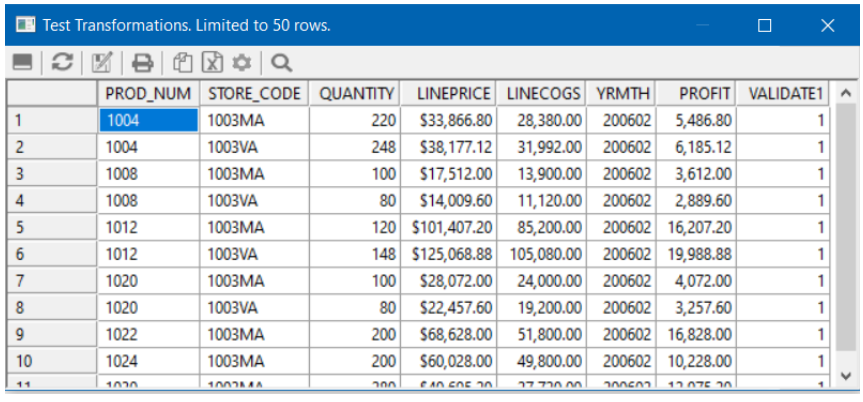
    Click the *Validates* tab in the Transformations dialog box. Then click the *Insert Intermediate Transforms* button to open the Validate Calculator.

11. To build the validation expression, double-click *QUANTITY* in the tree, and then select *GE* from the Relation drop-down menu. Input *10* into the Value field. Click *OK*. Your goal is to see which rows pass the validation test.

12. Now, let us test the validation and transformations to ensure that they are syntactically correct and performing the desired calculations. The test retrieves some rows from the server, applies the transformations, and displays the results.

    Click the *Test Transforms* button in the upper-right corner, above the Validates tab in the Transformations calculator.

    The result of the transformation test appears, as shown in the following image.



	PROD_NUM	STORE_CODE	QUANTITY	LINEPRICE	LINECOGS	YRMTH	PROFIT	VALIDATE1
1	1004	1003MA	220	$33,866.80	28,380.00	200602	5,486.80	1
2	1004	1003VA	248	$38,177.12	31,992.00	200602	6,185.12	1
3	1008	1003MA	100	$17,512.00	13,900.00	200602	3,612.00	1
4	1008	1003VA	80	$14,009.60	11,120.00	200602	2,889.60	1
5	1012	1003MA	120	$101,407.20	85,200.00	200602	16,207.20	1
6	1012	1003VA	148	$125,068.88	105,080.00	200602	19,988.88	1
7	1020	1003MA	100	$28,072.00	24,000.00	200602	4,072.00	1
8	1020	1003VA	80	$22,457.60	19,200.00	200602	3,257.60	1
9	1022	1003MA	200	$68,628.00	51,800.00	200602	16,828.00	1
10	1024	1003MA	200	$60,028.00	49,800.00	200602	10,228.00	1

    Note: The number of records retrieved will depend on the Run Options set from *Options*, in the Tools group on the Home tab, and on the database you choose for your sample data.

    In the VALIDATE1 column, the number 1 represents rows that will be accepted based on the validation test. The 0s represent rows that will be rejected because QUANTITY is less than 10.

    Tip: Since you have been working in the Transformations dialog box, testing from there is the simplest method, but you can also test the transformation by right-clicking the target object, clicking *Expand*, and clicking the Test Transforms tab.

13. Notice that Profits are positive in the answer set, which means that the transformation is working. It also satisfies the test condition, so you can move on to the next data target, which you will create from scratch.

14. Close the Test Transformation dialog box, and click *OK* to close the Transformations dialog box.

Create a Data Target

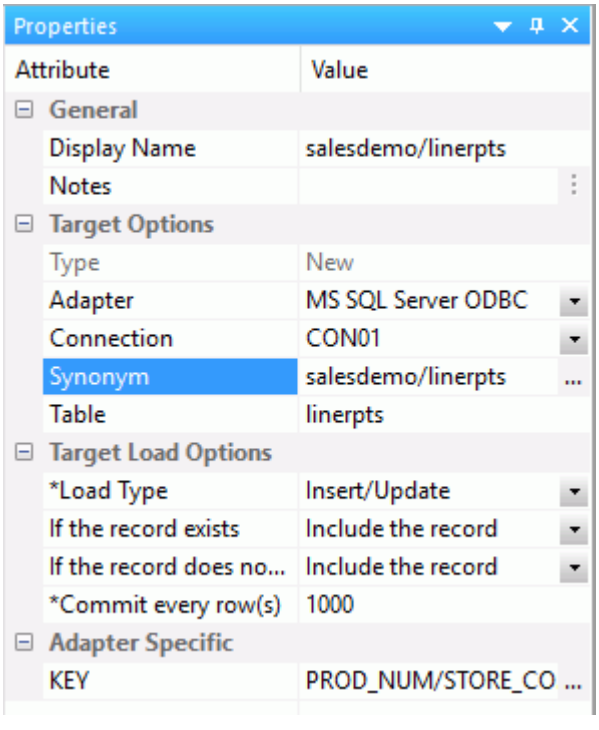
Data Migrator can load multiple data targets in a single data flow. Let us add another target object into the data flow. This time, the data target you need does not exist, so you will create it using the columns in the SQL SELECT statement that you defined for the SQL Select Columns object.

This target table will be used by line managers for competitive analysis. You will create the base table here, but other flows can update it with additional information.

1. Right-click in the workspace to the right of the SQL object, point to *Add Target*, and then click *New*. A new data target appears to the right of the SQL object.
2. Right-click the new target and click *Properties*.

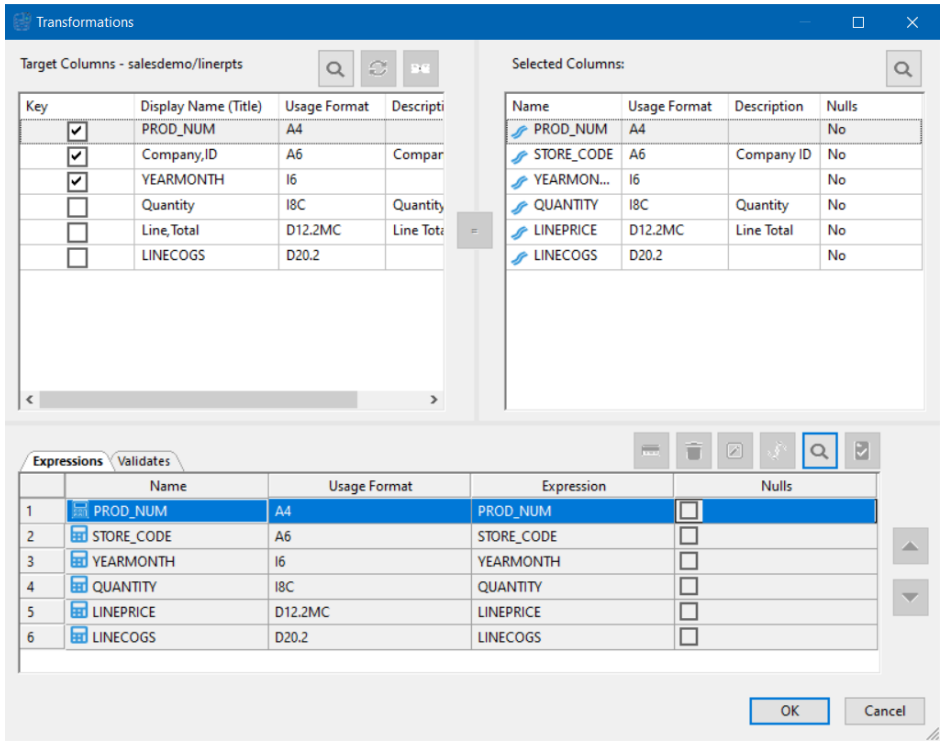
   The Target Properties pane opens.
3. To define target properties:
   a. Choose the same type of adapter you used for the data sources in our example. In this case, it was MS SQL Server ODBC.
   b. Change the value for Synonym to salesdemo/linerpts.
   c. Retain the default Load Type of Insert/Update.

      The target Properties pane should look like the following image:

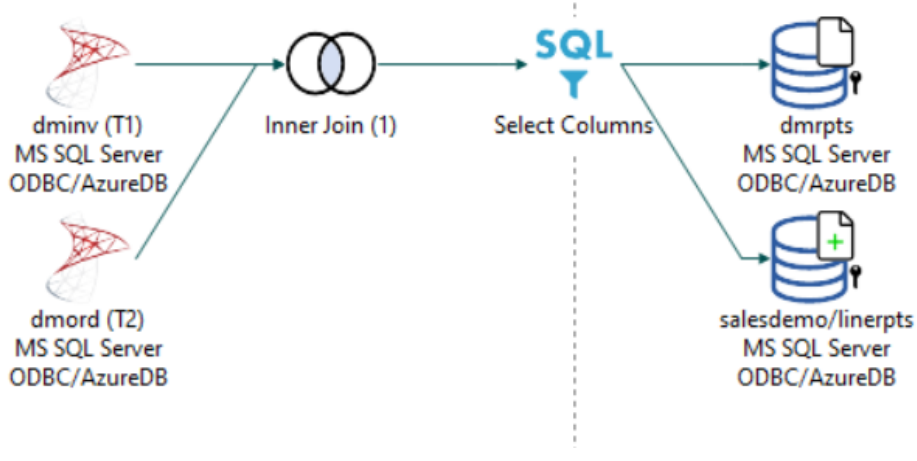


   d. Close the pane.
4. Right-click the new data target and click *Target Transformations*.

5. In the table you are creating, PROD_NUM, STORE_CODE, and YEARMONTH are all keys. Click each one to identify them, and those above it, as keys. The Transformations dialog box looks like the following image:



6. Click *OK* to return to the Data Flow tab, which now contains two data sources and two data targets, as shown in the following image.



Next Step: You are ready to save the data flow. Both data targets, dmrpts and linerpts, will be loaded based on your specifications.

Save and Name the Data Flow

Click the *Save* button in the Quick Access Toolbar.

The Save Procedure As window opens. Select which application directory to save the flow to using the Save In drop-down menu, and enter a flow name in the Name field. For this example, we will save our flow in the salesdemo application directory, and name it *salesdb*.

Next Step: You are ready to run the data flow.

Submitting the Data Flow

Now that you have completed the data flow, you can run it.

1. On the Home tab, in the Filter group, click *Procedures* to see all the procedures available. In the salesdemo application directory, right-click *salesdb* and click *Submit* from the menu.

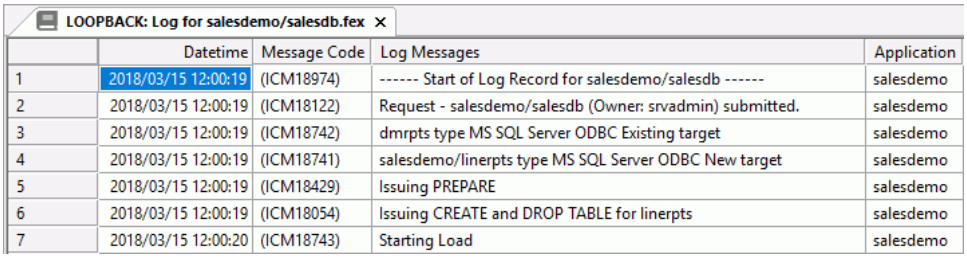
   or

   On the Flow tab, in the Run group, click *Run* and click *Submit* from the drop-down menu.

2. When the execution completes, you can review the log, which records the process and identifies any problems that may have occurred.

   From the navigation pane, right-click *salesdb* in the salesdemo application directory, point to *Logs*, and then click *Last Log*.

   The Log report opens in the workspace as shown in the following image.



	Datetime	Message Code	Log Messages	Application
1	2018/03/15 12:00:19	(ICM18974)	------ Start of Log Record for salesdemo/salesdb ------	salesdemo
2	2018/03/15 12:00:19	(ICM18122)	Request - salesdemo/salesdb (Owner: srvadmin) submitted.	salesdemo
3	2018/03/15 12:00:19	(ICM18742)	dmrpts type MS SQL Server ODBC Existing target	salesdemo
4	2018/03/15 12:00:19	(ICM18741)	salesdemo/linerpts type MS SQL Server ODBC New target	salesdemo
5	2018/03/15 12:00:19	(ICM18429)	Issuing PREPARE	salesdemo
6	2018/03/15 12:00:19	(ICM18054)	Issuing CREATE and DROP TABLE for linerpts	salesdemo
7	2018/03/15 12:00:20	(ICM18743)	Starting Load	salesdemo

3. Close the log window to return to the Data Flow tab.

   Notice that the Console Log indicates that the data flow has run successfully.

Data Flow Review

You have now completed Exercise 1, in which you built a data flow to move data out of the operational data store and into a denormalized table, so that the sales force can write meaningful sales reports, and the line managers can perform competitive analysis.

Next Step: When you are ready, proceed to Exercise 2 where you will create a process flow that uses the data flow you just built.

Exercise 2: Creating the Process Flow

A Process Flow controls how one or more data flows will be processed, by positioning a set of objects in the workspace and defining their interactions. You can combine:

- A Start object that defines when to run the process.
- Data Flow objects that indicate what data to extract, and copy from data sources to data targets.
- Email objects that notify users about the status of the process at specified points.
- Stored Procedure objects that perform a variety of supplementary tasks before or after the extraction, and load steps defined in a data flow.
- Connector objects, represented as arrows, that specify execution logic for the other objects included in the process flow. For example, what to do next when a step in the process succeeds or fails.
- Group objects that specify the flow of a subset of objects within the total flow. For example, two objects in a group that must be processed simultaneously, rather than sequentially, with the object that follows next. This could be an email notification that is waiting until processing has been completed for both of the grouped objects.

Scenario: You will create a process flow that uses the data flow you created in the last exercise, and then branches to one of two procedures based on the success or failure of the execution of that data flow. The appropriate results will be recorded in a log to facilitate quick trouble-shooting, if necessary, or immediate communication to sales agents and line managers that the latest data is available for reporting.

Tip: Although not demonstrated in the tutorial, you can extend the success and failure branches of the process flow to trigger the distribution of email messages, either to those in charge of correcting errors (upon failure), or to those who need the current data (upon success).

Next step: The simplest way to proceed would be to continue to work with *salesdb*. You could click the Process Flow tab to see the salesdb data flow object in the workspace, and then add other elements to create the process flow.

However, this is not necessarily the best way to proceed. Instead, you will use a method that is more *modular* and *flexible*. You will start a new flow, then click the Process Flow tab and drag the Data Flow object into the Process Flow workspace in the correct position in the flow. With this method, each flow is saved separately; a Data Flow object can then be used in more than one Process Flow, and a Process Flow can be modified as needed and used to incorporate different data flow objects.

Starting the Process Flow

Start the process flow just as you started the data flow:

1. Right-click the *salesdemo* directory, point to *New*, and then click *Flow*.

   A workspace opens in the right pane. The Data Flow tab is active.

2. Click the *Process Flow* tab to change your design mode.

   To anchor the Process flow, the Start object is automatically added to the workspace.

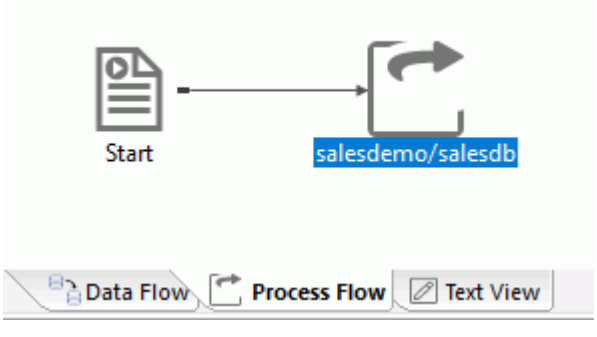
   Next step: You are going to add other objects to the flow and connect them to one another.

Adding a Data Flow Object to the Process Flow

To add the data flow *salesdb* to the process flow:

1. Drag the *salesdb* data flow from the navigation pane into the Process Flow workspace to the right of the *Start* object.
2. Right-click the *Start* object in the workspace and drag toward the *Data Flow* object, forming an arrow that connects the two objects.

   The Process Flow tab now looks like the following image:



Save and Name the Process Flow

The first time you save the process flow, you will assign a name to it.

1. Click the *Save* button. The Save Procedure As window opens.
2. Name the Process Flow *salesflow*.
3. Click *Save* to return to the Process Flow tab.

You will continue to save after each step.

Next step: Your next task is to create two stored procedures. You will connect these procedures to the data flow object so that one will run when the data flow executes successfully, and the other one will run when the data flow fails to execute properly.

Branching in the Process Flow

You want to be able to track success or failure when the data flow runs. You will do this by creating two procedures and including them in your process flow. Then you will supply the logic that invokes the appropriate one for the condition that is encountered.

Following the same modular model we have been using, you will create each of the stored procedures as a separate file that can be used in this process flow and others.

Create the Procedures

First, you will create a procedure that will appear in the log when the data flow executes correctly.

1. Right-click the *salesdemo* application directory, point to *New*, and then click *Procedure*.
2. A text editing window opens. Enter the following line of code:

   ```
   -TYPE SUCCESS!
   ```

3. Click the *Save As* button in the Quick Access Toolbar, and name this procedure *success*. Click *Save*.
4. Close the editing window.

   Now you will create a stored procedure that will appear in the log when the data flow fails to execute properly.

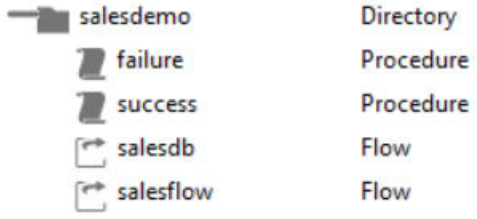
5. Right-click the *salesdemo* application directory again, point to *New*, and then click *Procedure*.
6. The text editing window opens. Enter the code:

   ```
   -TYPE FAILURE!
   ```

7. Click the *Save As* button in the Quick Access Toolbar, and name this procedure *failure*. Click *Save*.

   If applicable, this entry will be displayed in the Data Migrator log. Close the editing window.

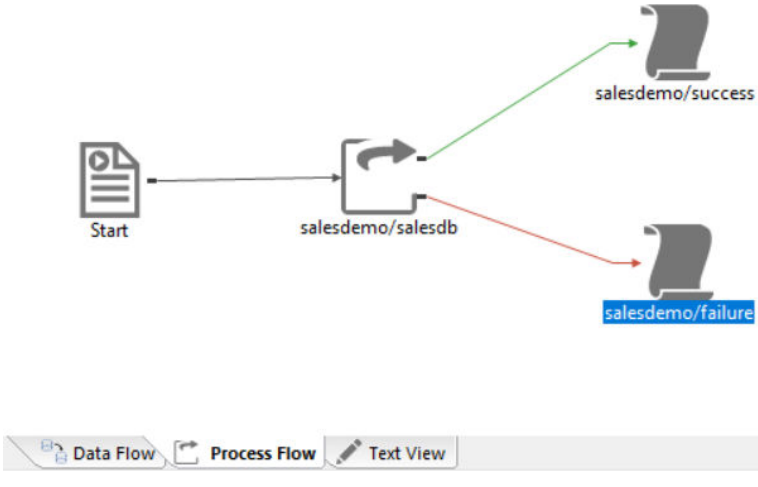
   Both procedures are now listed in the salesdemo application directory, as shown in the following image.



Connect the Procedures to the Data Flow

Drag the success procedure to the right of the data flow, and the failure procedure below the success procedure. Right-click the data flow and drag from it toe success. Then right-click the data flow and drag from it to the failure procedure.

The Process Flow tab should look like the following image:



Next step: You have now created two branches from the data flow object to the procedures. Now you need to define the behavior associated with each branch of the flow. You will do that from the connector arrows.

Specify Your Connection Logic

Follow-up actions based on the success or failure of the data flow execution are initiated by connection logic, in the form of return codes.

In the tutorial, we will use the default connection logic associated with the green and red arrows that branch to the stored procedures you created. Since these procedures write a line of code in the Data Migrator log, you will quickly be able to see which path the execution followed.

Tip: The implementation we have chosen to demonstrate can be built quickly and offers easily demonstrable results. However, there are many ways you can use branching within a process flow. For example:

- You might wish to send one email message in response to a return code associated with success, and a different message to different recipients in response to a return code associated with failure.

  or

- You might wish to initiate another data flow or process flow in response to a return code associated with success, but send an email message to someone charged with correcting processing errors in response to a return code associated with failure.

Your options are many and varied.

Let us now review the default logic that will trigger the running of your two stored procedures.

1. Double-click the *green arrow* connecting the data flow object to *success*. The Condition dialog box opens.

   OK Default Condition (Return Code is 0) is selected by default.

2. Click *OK*.
3. Now, double-click the red arrow that connects to the failed procedure. This time, *FAILED Default (Return Code is not 0)* is selected by default.
4. Click *OK*.

Next step: It is time to run the Process Flow to see the results as reflected in the Data Migrator log.

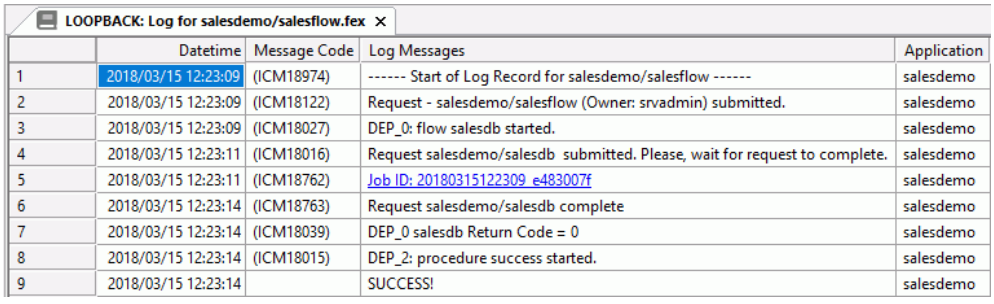
Submitting the Process Flow

To see if the process flow is successful, and if the proper messages are displayed, follow the steps below.

1. On the Flow tab, in the Run group, click *Run*, and choose *Submit* from the menu to test the flow.

2. From the navigation pane, right-click *salesflow* in the salesdemo application directory, point to *Logs*, and then click *Last Log*.

   The Log report opens in the workspace, as shown in the following image.



LOOPBACK: Log for salesdemo/salesflow.fex

	Datetime	Message Code	Log Messages	Application
1	2018/03/15 12:23:09	(ICM18974)	------ Start of Log Record for salesdemo/salesflow ------	salesdemo
2	2018/03/15 12:23:09	(ICM18122)	Request - salesdemo/salesflow (Owner: srvadmin) submitted.	salesdemo
3	2018/03/15 12:23:09	(ICM18027)	DEP_0: flow salesdb started.	salesdemo
4	2018/03/15 12:23:11	(ICM18016)	Request salesdemo/salesdb submitted. Please, wait for request to complete.	salesdemo
5	2018/03/15 12:23:11	(ICM18762)	Job ID: 20180315122309_e483007f	salesdemo
6	2018/03/15 12:23:14	(ICM18763)	Request salesdemo/salesdb complete	salesdemo
7	2018/03/15 12:23:14	(ICM18039)	DEP_0 salesdb Return Code = 0	salesdemo
8	2018/03/15 12:23:14	(ICM18015)	DEP_2: procedure success started.	salesdemo
9	2018/03/15 12:23:14		SUCCESS!	salesdemo

   The log shows the message

   SUCCESS!

   which was generated by the stored procedure called *success*.

3. If you click the highlighted line (Link to log), you will see the log of that step.
4. Close the log windows.

Congratulations. You have successfully completed Exercise 2.

You are now ready to create your own flows. Refer to *Designing a Data Flow* on page 171, and *Designing a Process Flow* on page 369, for detailed information about the options illustrated in this tutorial and many others that are available to you in the DMC.

Chapter 8

Generating Reports, Viewing the Log File, and Printing Flows

Data Migrator enables you to view reports on how your flows run and what error messages the Server generates. Impact Analysis reports allow you to assess component usage and evaluate the effect of changes. You can print a graphical representation of a flow from the workspace.

You can also run reports and view logs from the Web Console. For details, see the *TIBCO WebFOCUS® Reporting Server Administration* manual or the Web Console online help.

In addition to the reports described in this topic, Data Profiling information is available from the Synonym Editor. Data Lineage information is also available from the Synonym Editor. For information, see *Viewing Data Profiling Characteristics* on page 530, and *Viewing Data Lineage for a Column* on page 538.

In this chapter:

- Log, Statistics, and Summary Flow Reports
- Scheduler Reports
- Impact Analysis Reports on Synonyms, Procedures and Columns
- Working With a Report
- Viewing a Detailed Flow Report and Log File
- Printing a Graphical Representation of a Data or Process Flow

Log, Statistics, and Summary Flow Reports

The Data Migrator logs and traces folder contains a link that opens the Logs and Traces dialog box. This provides access to viewing two reports:

- The Log report contains the message output generated from running flows. This information includes the steps taken during data extraction, table creation, and data load, and any errors that occur. The report can include the entire log, or the log for a specific time period.

  For process flows, the Log report provides drill-down capabilities, which allow you to select entries that display details of the logs of the individual flows in the process flow.

- The Statistics report contains statistical information about flows, such as how many rows were written, when the flow ran, and how long it took. The report can include all statistics, or statistics over a specific time period, for all flows, or for different types of completed flows.

  The Statistics report is a drill-down report, which allows you to select entries that display details about the individual flows.

  Note: Statistics are not available for XML files described by multi-segment synonyms, or for flows that use load type Insert records from memory.

- The Summary Flow Report flow contains information about all of the flows in the server path or in a specific application directory. It includes the flow type, application directory, description, modify date, and creator.

Note:

- Data Migrator reports support flow names of up to 64 characters.
- The DM log and statistic tables are not updated with messages when flows are initiated using Run, from the Run group, or context menu.

Reference: Log Report Columns

Log reports provide the information listed and described in the following table:

Column	Function
Datetime	Date and time of log entry.
Message Code	Message code number.
Log Messages	Message strings in the log.
Application	Flow application directory.
Name	Flow name.
Job ID	Unique internal job name.

Note: If a log message contains a non-zero return code, the line will be shown in red.

Reference: **Statistics Report Columns**

Statistics reports provide the information listed and described in the following table:

Column	Function
User ID	User ID which executed a request.
Application	Flow application directory
Name	Flow name.
Log	Job ID as a link to the flow log.
Parent Log	Log for parent flow.
Start Datetime	Start date and time of execution.
End Datetime	End date and time of execution.
Total Runtime	Total run time for the flow.
Return Code	Internal return code for the flow.
Target Name	Target name. **Note:** This, and the remaining columns, apply on a per target basis.
Transactions	Number of rows processed.
Accept	Number of rows accepted.
Nomatch	Number of rows rejected.
DBMS Error	Number of rows rejected by the database for a constraint violation, such as a Unique Index constraint, a Not Null constraint, or a user-specified constraint. For a file target, the number of rows rejected due to format conversion errors.
Format	Number of format rejections.
Invalid	Number of rows rejected for failing validation tests.

Column	Function
Duplicates	Number of rejected duplicate records.
Input	Number of rows input.
Changed	Number of rows updated.
Deleted	Number of rows deleted.
Original Request	Original flow where dependency started.
Original User	Original user ID running dependent flow.
Dependency Count	Dependency number.
Message	Text explanation of error messages.
CPU Time	Total CPU time in milliseconds.

Procedure: How to Run a Log or Statistics Report

You can access Log and Statistics reports to view information about executed flows. The Log report displays the messages generated by the flows that have been run from the DMC, and the Statistics report provides information about the flows that have been scheduled or submitted from the DMC. The Statistics report also provides access to the message log of each flow.

1. From the navigation pane, expand the *Workspace* folder. Expand the *Logs and Traces* folder.
2. Double-click *Data Migrator Log and Statistics* or right-click it and click *View*.

   The Log and Statistics dialog box opens.
3. From the *Report Type* drop-down menu, select *Log* or *Statistics*.
4. Select values for the following fields and options to specify the information shown in the Log or Statistics report.

   Report Range

   Allows you to select what time range you want to run the report for. The options are Last, All, or Specify Time. If Specify Time is selected, an area opens where you can specify the start and end date and time. If Statistics is selected, the default value is All. If Log is selected, the default value is Last.

Completion Status

Allows you to view all reports, completed reports, reports completed successfully, or reports completed with errors. This option is only available when running a Statistics report.

Run by

Allows you to view reports for all flows, and those initiated by the scheduler or submit. This includes flows initiated from the DMC, WC, and CMRUN. This option is only available when running a Statistics report.

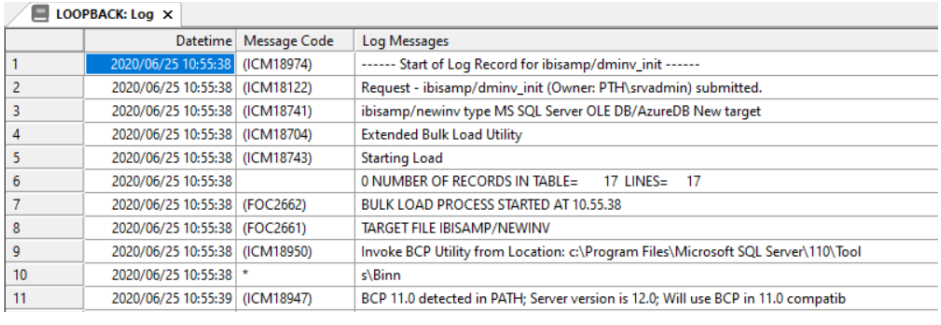
Application or Flow Name

Allows you to specify a specific application, flow, or procedure to run the report on.

5. Click *View Report*.

The report opens in the workspace.

An example of a Log report is shown in the following image.

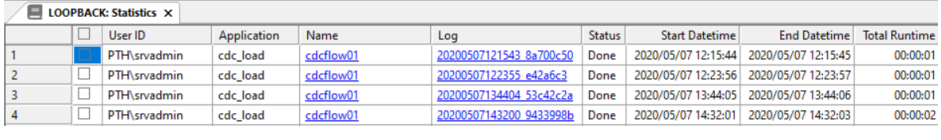


LOOPBACK: Log

	Datetime	Message Code	Log Messages
1	2020/06/25 10:55:38	(ICM18974)	------ Start of Log Record for ibisamp/dminv_init ------
2	2020/06/25 10:55:38	(ICM18122)	Request - ibisamp/dminv_init (Owner: PTH\srvadmin) submitted.
3	2020/06/25 10:55:38	(ICM18741)	ibisamp/newinv type MS SQL Server OLE DB/AzureDB New target
4	2020/06/25 10:55:38	(ICM18704)	Extended Bulk Load Utility
5	2020/06/25 10:55:38	(ICM18743)	Starting Load
6	2020/06/25 10:55:38		0 NUMBER OF RECORDS IN TABLE= 17 LINES= 17
7	2020/06/25 10:55:38	(FOC2662)	BULK LOAD PROCESS STARTED AT 10.55.38
8	2020/06/25 10:55:38	(FOC2661)	TARGET FILE IBISAMP/NEWINV
9	2020/06/25 10:55:38	(ICM18950)	Invoke BCP Utility from Location: c:\Program Files\Microsoft SQL Server\110\Tool
10	2020/06/25 10:55:38	*	s\Binn
11	2020/06/25 10:55:39	(ICM18947)	BCP 11.0 detected in PATH; Server version is 12.0; Will use BCP in 11.0 compatib

Note: For process flows, you can drill down to details of the flow components by double-clicking the blue Job ID line in a Log report.

An example of a Statistics report is shown in the following image



LOOPBACK: Statistics

	User ID	Application	Name	Log	Status	Start Datetime	End Datetime	Total Runtime
1	PTH\srvadmin	cdc_load	cdcflow01	20200507121543_8a700c50	Done	2020/05/07 12:15:44	2020/05/07 12:15:45	00:00:01
2	PTH\srvadmin	cdc_load	cdcflow01	20200507122355_e42a6c3	Done	2020/05/07 12:23:56	2020/05/07 12:23:57	00:00:01
3	PTH\srvadmin	cdc_load	cdcflow01	20200507134404_53c42c2a	Done	2020/05/07 13:44:05	2020/05/07 13:44:06	00:00:01
4	PTH\srvadmin	cdc_load	cdcflow01	20200507143200_9433998b	Done	2020/05/07 14:32:01	2020/05/07 14:32:03	00:00:02

To view the message log for an individual flow, double-click its link or right-click the link and click *Show Log* in a Statistics report. If the log has output, right-click the log, and click *Output*. Failed flows will be shown in red.

To delete a log and its output, select its corresponding check box in a Statistics report, and on the Report tab, in the Tools group, click *Delete Selected*.

Procedure: How to Run a Summary Flow Report

A Summary Flow Report displays all of the flows on the server or in a specific application directory.

In the navigation pane, right-click a specific application directory folder. Point to *Impact Analysis*, then *Flow Report*, and then click *Summary*.

The report opens in the workspace, as shown in the following image.

LOOPBACK: Flow Summary for dmgeneral

dmgeneral	dmodate	Procedure	Set global variable ODATE for use in dmvar data flow
dmgeneral	dmordxml	Direct Load Flow	Target (XML) described by dmxord.xsd
dmgeneral	dmorecs	Flow	Rectype type used to create one file with two record types
dmgeneral	dmpeople	Flow	Use UNION to combine two select statements
dmgeneral	dmpivot	Flow	Rectype used to pivot file with two record types
dmgeneral	dmquant	Flow	Nested select join to subselect with aggregate
dmgeneral	dmrctyp	Flow	Rectype used to create flat file with two record types
dmgeneral	dmreuse	Flow	Defined Functions used
dmgeneral	dmscini	Flow	Sales Counts Initial Load
dmgeneral	dmscupd	Flow	Sales Counts Updates using Match Expressions
dmgeneral	dmseq	Flow	Target (flat file) with packed decimal data
dmgeneral	dmsplit	Flow	target (multiple tables) - validate used to split into two tables
dmgeneral	dmstarterm	Direct Load Flow	Direct load from file with multiple record types
dmgeneral	dmstdef	Procedure	FILEDEF issued for decode file
dmgeneral	dmsubsel	Flow	Sub selects in select list
dmgeneral	dmsurrg	Flow	Surrogate key - generation using compute using compute

Scheduler Reports

The Scheduler reports folder include two reports:

- The Scheduler Log report displays the date/time when scheduled flows were submitted, as well as their Job IDs. The report can include the entire log, or the log for a specific time period.

  The Scheduler Log report is a drill-down report, which allows you to click entries to display details about the individual flows.

- The Scheduled Events report shows what flows are scheduled to run on a specified day. The default is to display information about the current date.

Procedure: How to Run a Scheduler Log Report

The Scheduler Log report displays the date/time when scheduled flows were submitted, as well as their Job IDs.

1. From the navigation pane, expand the *Workspace* folder. Expand the *Logs and Traces* folder.
2. Right-click *Data Migrator Log and Statistics* and click *View*.

   The Log and Statistics dialog box opens.
3. For Report Type, select *Statistics*, for Report Range, select *All*, and for Run by, select *Scheduler*.
4. Click *View Report*.

The report of scheduled flows opens.

Note: Double-clicking a Link to log row will display the log for that flow.

Reference: Scheduler Log Report Columns

Scheduler Log reports provide the information listed and described in the following table:

Column	Format	Function
SUBMIT_DATE_TIME	Alpha 20	Submit date and time for flow.
APPNAME	Alpha 64	Flow application directory
REQUEST	Alpha 64	Flow name.
LOG	Alpha 64	Link to the flow log.

Procedure: How to Run a Scheduled Events Report

The Scheduled Events reports provides information about what time flows are scheduled to run on a particular day.

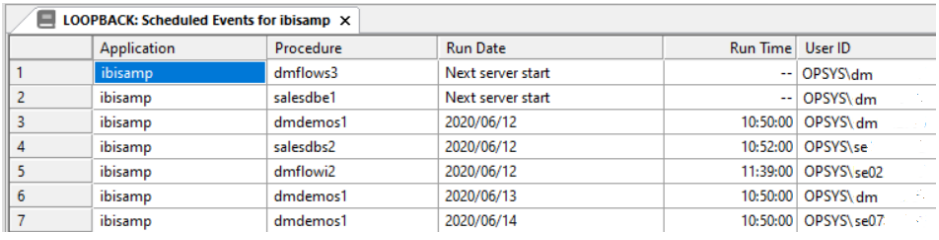
1. In the navigation pane, right-click an application directory, point to *Schedule and E-mail*, and then click *Scheduled Events*.

   The Scheduled Events window opens.
2. The report defaults to today. Optionally, you can select a date and time in the future.

Note: The date-field drop-down menu launches a calendar for selecting dates. The time-field scroll buttons allow you change the highlighted section of the time. For example, you can change the hour, minute, or second.

3. Click *View Report*.

The report opens in the workspace, as shown in the following image.



LOOPBACK: Scheduled Events for ibisamp

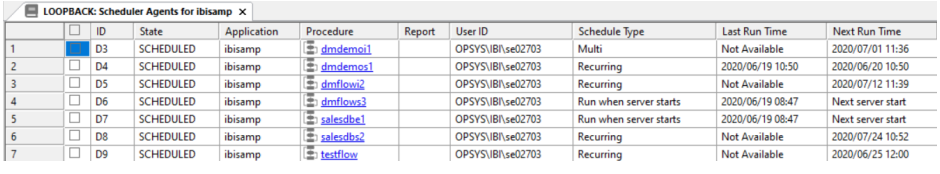
	Application	Procedure	Run Date	Run Time	User ID
1	ibisamp	dmflows3	Next server start	--	OPSYS\dm
2	ibisamp	salesdbe1	Next server start	--	OPSYS\dm
3	ibisamp	dmdemos1	2020/06/12	10:50:00	OPSYS\dm
4	ibisamp	salesdbs2	2020/06/12	10:52:00	OPSYS\se
5	ibisamp	dmflowi2	2020/06/12	11:39:00	OPSYS\se02
6	ibisamp	dmdemos1	2020/06/13	10:50:00	OPSYS\dm
7	ibisamp	dmdemos1	2020/06/14	10:50:00	OPSYS\se07

Procedure: How to Run a Scheduler Agents Report

The Scheduler Agents report provides information about every scheduled flow and the next and last runs.

In the navigation pane, right-click an application directory, point to *Schedule and E-mail*, and then click *Scheduler Agents*.

The Scheduler Agents report opens, as shown in the following image.



LOOPBACK: Scheduler Agents for ibisamp

	ID	State	Application	Procedure	Report	User ID	Schedule Type	Last Run Time	Next Run Time
1	D3	SCHEDULED	ibisamp	dmdemoi1		OPSYS\IBI\se02703	Multi	Not Available	2020/07/01 11:36
2	D4	SCHEDULED	ibisamp	dmdemos1		OPSYS\IBI\se02703	Recurring	2020/06/19 10:50	2020/06/20 10:50
3	D5	SCHEDULED	ibisamp	dmflowi2		OPSYS\IBI\se02703	Recurring	Not Available	2020/07/12 11:39
4	D6	SCHEDULED	ibisamp	dmflows3		OPSYS\IBI\se02703	Run when server starts	2020/06/19 08:47	Next server start
5	D7	SCHEDULED	ibisamp	salesdbe1		OPSYS\IBI\se02703	Run when server starts	2020/06/19 08:47	Next server start
6	D8	SCHEDULED	ibisamp	salesdbs2		OPSYS\IBI\se02703	Recurring	Not Available	2020/07/24 10:52
7	D9	SCHEDULED	ibisamp	testflow		OPSYS\IBI\se02703	Recurring	Not Available	2020/06/25 12:00

Reference: Scheduler Agents Report Columns

Scheduled Events reports provide the information listed and described in the following table:

Column	Function
ID	The identification number associated with the scheduler agent.
State	The current state of the scheduler agent.
Application	The application name of the scheduled procedure.

Column	Function
Procedure	The name of the scheduled procedure.
Report	An optional report name.
User ID	The user credentials used to run the scheduled procedure.
Schedule Type	How often the procedure is run.
Last Run Time	The last time the procedure ran.
Next Run Time	The time of the next scheduled run.

Reference: Scheduled Events Dialog Box

To access the Scheduled Events dialog box from the DMC, expand the *Workspace* folder. Then, expand the *Special Services and Listeners* folder, right-click *Scheduler* and click *Scheduled Events*. You can also right-click an application directory, point to *Schedule and E-Mail*, and then click *Scheduled Events*.

The DM Scheduled Events window has the following fields and options:

Start

Start of the specific time period for generating this report.

Note: The date-field drop-down menu launches a calendar for selecting dates. The time-field scroll buttons allow you change the highlighted section of the time. For example, you can change the hour, minute, or second.

End

End of the specific time period for generating this report.

Note: The date-field drop-down menu launches a calendar for selecting dates. The time-field scroll buttons allow you change the highlighted section of the time. For example, you can change the hour, minute, or second.

Impact Analysis Reports on Synonyms, Procedures and Columns

Note: Both flows and stored procedures are referred to generically as procedures.

Impact Analysis Reports on procedures, synonyms, and columns provide information on the flows and stored procedures available on a particular server, and the synonyms and columns they use. They allow you to evaluate changes before they are made by showing which components will be affected. Impact Analysis Reports provide quick analysis of synonym and column usage, and allow you to get procedure information without opening them individually. These reports are drill-down reports, which allow you to double-click entries to open the procedures in the workspace. Impact Analysis reports are also available for individual synonyms and columns.

There are six types of reports:

- **Synonyms by Procedure report.** Lists your procedures and the synonyms that they use. This list is sorted by procedure. The report includes whether the synonym is used as a source, a target, or iterator, the adapter type, the table name and description, load type and prior to load options, and directory locations.
- **Procedures by Synonym report.** Lists your synonyms and the procedures that use them. This list is sorted by synonym. The report includes whether the synonym is used as a source, a target, or iterator, the adapter type, the table name and description, load type and prior to load options, and directory locations.
- **Columns by Procedure report.** Lists the columns used in your procedures and their synonyms. This list is sorted by data flow. The report includes how the column is used, as well as directory locations.
- **Procedures by Column report.** Lists your procedures and their synonyms. This list is sorted by column. The report includes how the column is used, as well as directory locations.
- **Synonyms by Procedure - Enterprise report.** Includes the same information as a Synonyms by Procedure report, but for the server and all its subservers in a hub or subserver environment. This report does not appear if the server you are reporting on does not have any remote servers configured.
- **Procedures by Synonym - Enterprise report.** Includes the same information as a Procedure by Synonym report, but for the server and all its subservers in a hub or subserver environment. This report does not appear if the server you are reporting on does not have any remote servers configured.

Note: Enterprise reports do not appear if you do not have any subservers.

Procedure: How to Run an Impact Analysis Report on Synonyms, Procedures, and Columns

Note: Both flows and stored procedures are referred to generically as procedures.

Impact Analysis reports provide information on the procedures, synonyms, and columns available on a particular server. They allow you to evaluate changes before they are made, by showing which components will be affected. The reports can be sorted by procedure, by synonym, or by column. The results can be exported to Microsoft Excel, copied as text, or printed. For details, see *Working With a Report* on page 465.

1. In the navigation pane, right-click a specific application directory, and point to *Impact Analysis*.
2. From the shortcut menu, click *Synonyms by Procedure*, *Synonyms by Procedure for Enterprise, Procedures by Synonym*, *Procedures by Synonym for Enterprise, Columns by Procedure*, or *Procedures by Column*. You can also point to *Flow Report* and click a *Summary* or *Detailed* report.

   The report opens in the workspace.

These reports include the procedure and synonym directories, columns and column types, synonym usage (source or target), adapter type, table name, description, and prior to load and load type options. These reports are drill-down reports, which allow you to double-click the procedure name entries to open the flows or stored procedures in the workspace.

The Synonyms by Procedure report shown in the following image lists each procedure and the synonyms it uses. The Synonyms by Procedure for Enterprise report shows the same report for the application directory and its subdirectories.

LOOPBACK: Synonyms by Procedure for ibisamp

	Found In Application	Found In	Found In Type	Synonym Application	Synonym	Usage
1	ibisamp	cargraph	Procedure		&table	GRAPH
2	ibisamp	cargraph	Procedure		&table	TABLE
3	ibisamp	carinst	Procedure	ibisamp	car	TABLE
4	ibisamp	carinst2	Procedure	ibisamp	car	TABLE
5	ibisamp	carmgn	Procedure	ibisamp	car	DEFINE
6	ibisamp	carmgn	Procedure	ibisamp	car	TABLE
7	ibisamp	courses	Procedure	ibisamp	courses	CREATE

The Procedures by Synonym report shown in the following image lists each synonym and the procedures that use it. The Procedures by Synonym for Enterprise report shows the same report for the application directory and its subdirectories.

LOOPBACK: Procedures by Synonym for ibisamp

	Synonym Application	Synonym	Found In Application	Found In	Found In Type	Usage
1		&table	ibisamp	cargraph	Procedure	GRAPH
2		&table	ibisamp	cargraph	Procedure	TABLE
3	ibisamp	car	ibisamp	carinst	Procedure	TABLE
4	ibisamp	car	ibisamp	carinst2	Procedure	TABLE
5	ibisamp	car	ibisamp	carmgn	Procedure	DEFINE
6	ibisamp	car	ibisamp	carmgn	Procedure	TABLE
7	ibisamp	courses	ibisamp	courses	Procedure	CREATE

The Columns by Procedure report shown in the following image lists each procedure and the synonyms and columns it uses.

LOOPBACK: Columns by Procedure for ibisamp

	Found In Application	Found In	Found In Type	Synonym Application	Synonym	Column
126	ibisamp/se	carbv	Business View	ibisamp/se	carbv	DEALER_COST
127	ibisamp/se	carbv	Business View	ibisamp/se	carbv	CAR
128	ibisamp/se	carbv	Business View	ibisamp/se	carbv	SALES
129	ibisamp/se	carbv	Business View	ibisamp/se	carbv	MARKUP
130	ibisamp	carinst	Procedure	ibisamp	car	CAR
131	ibisamp	carinst	Procedure	ibisamp	car	MODEL
132	ibisamp	carinst	Procedure	ibisamp	car	SEATS

The Procedures by Column report shown in the following image lists each column by synonym and by the procedures that use it.

LOOPBACK: Procedures by Column for citibike

	Column	Usage	Line	Used In	Identifier	Synonym Application
1	TRIPDURATION	Business View Reference	112		TRIPDURATION	citibike
2	START_STATION_ID	Business View Reference	113		START_STATION_ID	citibike
3	END_STATION_ID	Business View Reference	114		END_STATION_ID	citibike
4	BIKEID	Business View Reference	115		BIKEID	citibike
5	BIRTH_YEAR	Business View Reference	116		BIRTH_YEAR	citibike
6	GENDER	Business View Reference	117		GENDER	citibike
7	STARTTIME	Business View Reference	123		STARTTIME	citibike

Procedure: How to Run an Impact Analysis Report on a Particular Synonym

An Impact Analysis report on a particular synonym provides information on the procedures and reports that use it and how it is used. It allows you to evaluate changes to synonyms before they are made, by showing which components will be affected. The results can be exported to Microsoft Excel, copied as text, or printed.

These reports are drill-down reports, which allow you to double-click entries to open procedures in the workspace to make the necessary changes.

1. Right-click a synonym in the navigation pane and click *Impact Analysis*.

The Impact Analysis report opens in the workspace.

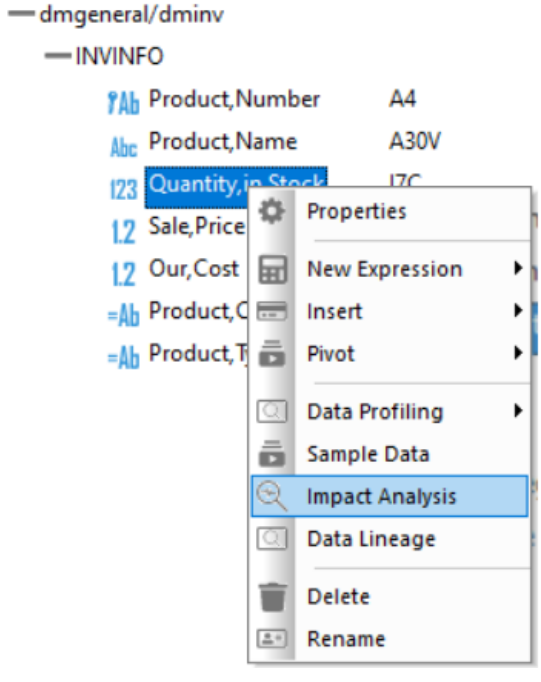
2. Double-click a procedure name to open it in the workspace.

Procedure: How to Run an Impact Analysis Report on a Particular Column

An Impact Analysis report on a particular column provides information on the procedures and reports that use it and how it is used. It allows you to evaluate changes to the column before they are made, by showing which components will be affected. The results can be exported to Microsoft Excel, copied as text, or printed.

These reports are drill-down reports, which allow you to double-click entries to open procedures in the workspace to make the necessary changes.

1. Double-click a synonym in the navigation pane to open it in the Synonym Editor.
2. Right-click a column in the Synonym Editor and click *Impact Analysis*, as shown in the following image.



The Impact Analysis report opens in the workspace.

3. Double-click a procedure name to open it in the workspace.

Procedure: How to Run an Impact Analysis Report on a Particular Procedure

An Impact Analysis report on a particular procedure provides information on that procedure, reports that use it, and how it is used. It allows you to evaluate changes to the procedure before they are made, by showing which components will be affected. The results can be exported to Microsoft Excel, copied as text, or printed.

These reports are drill-down reports, which allow you to double-click entries to open procedures in the workspace to make the necessary changes.

1. Right-click a procedure in the navigation pane, point to *Analysis Reports*, and click *Impact Analysis*.

   The Impact Analysis report opens.

2. Double-click a procedure name to open it in the workspace.

   Note: The Analysis Reports folder also gives you access to the Dependency Analysis and Data Lineage reports.

Reference: Synonym, Data Flow, and Column Impact Analysis Report Columns

Synonym, Procedure, and Column reports provide the information listed and described in the following table:

Column	Format	Function
Found In	Alpha 32	Flow or stored procedure name.
Found In Application	Alpha 18	Flow or stored procedure application directory.
Found In Type		Indicates if the object was found in a stored procedure or flow.
Synonym	Alpha 64	Synonym name.
Synonym App	Alpha 64	Synonym application directory. **Note:** This value will only appear if you are using application directories.
Usage	Alpha 36	How the synonym or column is used, for example, the source, target, or transformation.
Line	Alpha 8	The line number where the item appears.

Column	Format	Function
Adapter	Alpha 64	Synonym format.
Table Name or Dataset	Alpha 133	Data target table name.
Synonym Remarks	Alpha 78	Synonym description.
Prior to Load	Alpha 13	Prior to load option for targets.
Load Type	Alpha 35	Load type for targets.
Procedure Description	Alpha 150	Flow or stored procedure description.
Column	Alpha 66	Column name.
Used In	Alpha 100	The expression that uses a particular column.
Identifier	Alpha 66	The expression unique identifier.

Working With a Report

You can use report data by copying it into another application, or by exporting it into Microsoft Excel. You can also manipulate the display of a report and print it.

For information about the Reports toolbar, which becomes active when you run a report, see *Reports Tab* on page 140.

Procedure: How to Copy Report Output to the Clipboard

You can copy all or part of a report to the clipboard and paste it into a text editor or other application.

After running a report:

1. Select the part of the report that you want to copy. To copy the entire report, click the upper left hand cell, which selects all rows and columns.

   Note: Select a row by clicking its row number. Select a column by clicking its title. Use the Shift and Ctrl keys to make multiple row or column selections.

2. On the Report tab, in the Tools group, click *Copy as Text*, or right-click the selected text and click *Copy as Text*.

3. In another application, select *Paste* from the Edit menu or use Ctrl+V to copy the selected data.

Procedure: How to Export a Report to Microsoft Excel

You can export all or part of a report to Microsoft Excel.

Note: This option is only available with the Alpha or Original output format.

After running a report:

1. Select the part of the report that you want to export. If you do not select a part of the report, the entire report will be exported.

   Note: Select a row by clicking its row number. Select a column by clicking its title. Use the Shift and Ctrl keys to make multiple row or column selections.

2. On the Report tab, in the Tools group, click *Export to Excel*, or right-click the selected text and click *Export to Excel*.

The report opens in Microsoft Excel.

Procedure: How to Change the Report Font

You can change the font used in reports.

1. On the Home tab, in the Tools group, click *Options*.

   The Options dialog box opens.

2. Click the *Format* link on the User Preferences tree.

   The Format settings page opens in the right pane.

3. Click *Data Migrator Reports* in the *Fonts* text box.
4. Click the *Font* button.

   The Font dialog box opens.

5. Select the font attributes, and click *OK*.

The changes will be applied the next time you run a report.

Procedure: How to Change the Report Format

You can change the format for reports.

1. On the Home tab, in the Tools group, click *Options*.

   The Options dialog box opens.

2. Click the *Run Options* link on the User Preferences tree.

The Run Options settings page appears in the right pane.

3. From the Test reports default format drop-down menu, select which format you would like the report to be in.

   The format options are the following:

   - **HTML**
   - **HTML - plain text**
   - **active report**
   - **Excel**
   - **PDF**
   - **Unformatted**

   For more information regarding setting preferences for test reports, see *User Preferences - Run Options* on page 166.

Procedure: How to View the Top Level Procedure

When you schedule or submit a process flow that calls other flows, it can be useful to see both the name of the flow that is currently running, and the one that started the process while viewing the Data Services Agents report.

To view a top level procedure:

1. From the DMC, with an Application or Server Administrator user ID, expand the *Workspace*folder.
2. Right-click *Data Services* and click *Agents*.
3. On the Report tab, in the Tools group, click *View* and then click *Choose Columns*.

   The Select the columns dialog box opens.
4. Select the *Top Level Procedure* check box from the Category/Title column.
5. Click *OK*.
6. Click the *Refresh* button.

As shown in the following image, the dsloadfact procedure was called by the Top Level Procedure ibisamp/dsflows.

mid	Service	State	Scheduler Job ID	Query Time	Command	Procedure	Top Level Procedure
10	DEFAULT	idle	--	--	--	--	--
11	WC_DEFAULT	idle	20111229111222_f272ef2a	Dec 29 11:12:35	SHH	dsloadfact	ibisamp/dsflows

Procedure: How to Print a Report

When viewing a report on the screen, you can print the report by using any of the following methods:

- Clicking the *Print*  button from the Quick Access Toolbar.
- Right-clicking anywhere in the report and selecting *Print* from the context menu.
- Pressing Ctrl+P.

The standard Print dialog box opens, where you can select a printer destination, the number of copies, and other print properties.

Note: If you selected a range of cells in the report, you can click the *Selection* radio button to only print that range.

Procedure: How to Print Preview a Report

When viewing a report on the screen, you can preview the report by using any of the following methods:

- Clicking the *Print Preview* button from the Quick Access Toolbar.
- Right-clicking anywhere in the report and selecting *Print Preview* from the context menu.
- Pressing Ctrl+I.

The print preview window opens in the workspace.

Reference: Print Preview Tab

The Print Preview tab contains groups and commands that allow you to see a preview of the report that you are going to print. The Print Preview tab contains the following groups:

Print Group

The Print group consists of the following command.

Print

Prints the report.

Zoom Group

The Zoom group consists of the following commands.

Zoom In

Zooms in on the page.

Zoom Out

Zooms out of the page.

One Page/Two Page

Switches the view between one or two pages.

Preview Group

The Preview group consists of the following commands.

Next Page

Shows the next page of the report.

Previous Page

Shows the previous page of the report.

Close

Closes the print preview.

Procedure: How to Print a Report for Specific Pages

When viewing a report on the screen:

1. Select *Print* from the Quick Access Toolbar.

   The Print dialog box opens.

2. Select the *Pages* radio button in the Print Range section, and enter the page numbers that you want to print.

3. Click *OK*.

Procedure: How to Print a Report for Selected Rows or Columns

When viewing a report on the screen in a grid:

1. Select the columns you want to print by clicking on the title bar. You can also select specific rows and columns by pressing the Ctrl or Shift key.
2. Select *Print* from the Quick Access Toolbar.

   The Print dialog box opens.
3. Select the *Selection* radio button from the Print Range section.
4. Click *OK*.

Viewing a Detailed Flow Report and Log File

A detailed flow report displays general properties like name, description, DM release, creator, and modify date. It also provides specifics on SQL Select Statements, target information, load options, target transformations, and execution properties. These reports are useful for documentation purposes.

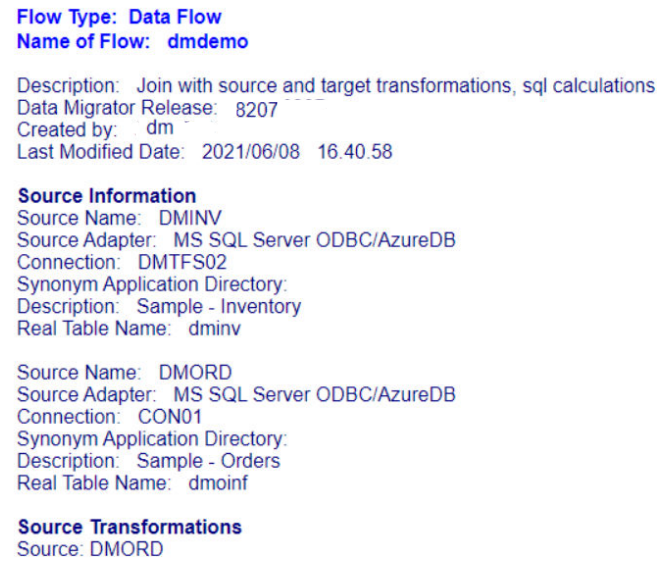
Detailed reports are available for individual flows from the DMC. You also can generate detailed reports for all of the flows in the path of the server, or in a specific application directory from the Web Console.

The log file for a flow is useful in determining problems encountered when running a flow.

Procedure: How to View a Detailed Flow Report

Right-click a flow in the navigation pane and click *Flow Report*.

The flow report shown in the following image appears in the workspace. It can be printed from that window.



Tip 1: You can also copy the report to the Clipboard and paste it in Word or any other word processing program. To do this, right-click the report and click *Select All*. Then right-click and select *Copy*. You can now paste the report into another program.

Procedure: How to View and Print a Set of Detailed Flow Reports

You can generate detailed reports for all of the flows in the server path or in a specific application directory.

Note: If there are many flows in this report, it may take a while to generate.

1. Right-click the server or an application directory, select *Impact Analysis*, and then click *Flow Report*.
2. Select *Detailed* from the context menu.

For output in default format, the first report is displayed. On the Report tab, in the Run group, click *Next Result* to view the next report, and continue until you have viewed all the reports.

For pdf output, a single multi-page document is displayed.

The flow reports appear in the workspace.

Note: The *Next Result* common in the Run group, on the Report tab, lets you navigate to the next flow report.

Procedure: How to View the Log File

You can generate a log file for reports that have been run.

1. Right-click a flow in the navigation pane, select *Logs*, and then click *Log and Statistics*.

   The Log and Statistics dialog box opens.

2. From the Report Type drop-down menu, select *Log*.

   Specify what time range you want to run the report for. The options are Last, All, or Specify Time.

3. Select *View Report*.

   The log report appears in the workspace.

Example: Sample Data Migrator Log File

This example contains sample lines from log files, and an explanation of the code.

The first lines of a log display DM job initialization.

- This request was submitted:

```
------ Start of Log Record for baseapp/one ------
Request - one (Owner: dm) submitted.
```

The next lines display the data target or file type, and whether it is a new or existing data target.

- The following is a new table, and Data Migrator attempts to drop and recreate the target table:

```
linertps type MS SQL Server New target
PREPARE executed
Create synonym executed for: linertps
Issuing DROP TABLE for linertps
Issuing CREATE TABLE for linertps
```

- The following is an existing table:

```
dmrpts type MS SQL Server Existing target
PREPARE executed
ROWS AFFECTED BY PASSTHRU COMMAND  : 222/DELETE
```

The log then displays summary data for the request. You can see the number of records that were processed.

- In this example, there were 200 rows processed, which Data Migrator attempted to insert into the target table. Of those, 20 were rejected due to duplicate keys, and 180 were accepted by the data target.

```
Starting Load-- stats for newtab -------------------------------------
184 : Row(s) processed by job
144 : Row(s) accepted by target table
144 : Row(s) inserted into target table
0 : Row(s) updated in target table
0 : Row(s) deleted from target table
0 : Row(s) rejected due to format errors
40 : Row(s) rejected due to validation errors
0 : Row(s) rejected due to no match logic
0 : Row(s) rejected because duplicate exists
```

The remaining lines contain summary data and the return code from the flow.

- In the following, a return code of zero (0) means the request completed successfully:

```
Ending Load
Return Code = 0
Request: one - finished processing
Finished
```

Printing a Graphical Representation of a Data or Process Flow

As a flow gets more complex, it can be helpful to print out a graphical representation. This printed version of the flow can facilitate analysis and assist in troubleshooting. Flows are printed from the application that is associated with bitmap files on your desktop.

Procedure: How to Print a Graphical Representation of a Flow

1. Open the flow in the workspace.
2. Click the Print button on the Quick Access Toolbar.

   An image of the flow opens in the application that is associated with bitmap files on your desktop.
3. Select *Print* from the application File menu.

Note: If no application is associated with bitmap files, the graphical representation will open in the default Microsoft Picture and Fax Viewer.

Chapter 9

Optimizing Performance

The run-time performance of Data Migrator depends on many factors: host platform, network protocol, I/O devices, type of SQL command, and selected DBMS. Depending on your environment, one or more of these factors may be in your control. The following topics outline the options that have the greatest impact on performance.

The issues described here may or may not affect your particular configuration. After careful review and comparison, you may determine that your request will need a lot of internal processing and significant I/O. This may be fine if your nightly incremental updates are low in volume or if you are on a processor where disk access is not a bottleneck. However, if you have only one hour to process your data and the server is on a small machine with few resources, an alternate approach may be necessary.

In either case, review these performance considerations, and compare them to your environment. For most applications, the default functionality of Data Migrator is sufficient for your data management needs. If the performance objectives cannot be met, these guidelines can at least provide you with some insight into the location of potential bottlenecks and possible strategies to overcome them.

In this chapter:

- Improving Data Extraction Performance
- Using Automatic Passthru
- Improving Data Loading Performance
- Improving Lookup Performance
- Bulk Retrieval With Salesforce.com

Improving Data Extraction Performance

Data Migrator extracts data from data sources using SQL. The process of applying SQL to heterogeneous data sources is very complex and too broad a subject to easily outline here. However, the following are issues which you control.

- **Providing qualifiers to the data source.** Data Migrator uses index structures (if available) to retrieve rows in the most efficient manner. Therefore, it is advisable to supply screening conditions in your WHERE criteria for as many columns as possible.

Make sure that critical native DBMS optimization fields (primary keys, secondary indexes, and so on) are correctly defined in the synonym. For more information on defining synonyms, see Chapter 5, *Metadata*, in the *TIBCO WebFOCUS® Reporting Server Administration* manual.

- **Evaluating your joins.** Data Migrator is flexible, enabling similar data source, cross-data source, and even cross-server (sometimes called cross-platform) joins. The trade off for this flexibility is the amount of work required of the server, and the number of intermediate files being created. To minimize the impact of joins, try the following:
  - Join your tables using key columns and indexes.
  - Minimize the number of tables used in a single SELECT command.
  - Specify the *smaller* of the two tables being joined first.

Using Automatic Passthru

Automatic Passthru (APT) is the ability of a Reporting Server to pass or hand-off an intact SQL command to a remote server or RDBMS subsystem. In effect, it enables a user to speed up Data Migrator. If a request does not meet the requirements for APT, the Reporting Server needs to do additional work. SQL commands must be translated into a format (Data Manipulation Language) that is understood by the underlying (typically non-relational) file system(s), forcing the need for a special work file to be created on the Server platform. Data Migrator waits for this temporary work file to be written, and then continues processing by loading the RDBMS table.

If a request meets the requirements for APT, Data Migrator uses its streaming or pipeline methodology. This means no intermediate files are written by Data Migrator. Rows are fetched directly from the source (effectively off the wire for communication scenarios) and are then written to the target. Whether Data Migrator is writing a local fixed format file, or sending dynamic SQL to an RDBMS table, an SQL request that runs in APT mode guarantees the best default performance.

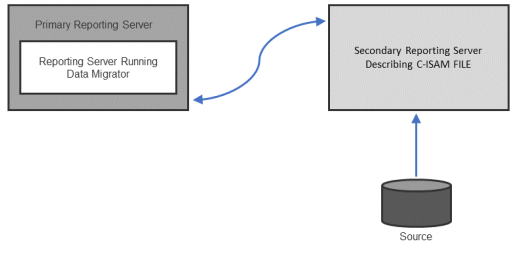
The following environmental and request-based factors most often result in requests failing the requirements of APT:

- Non-relational file structures accessed directly at the server.
- DEFINE commands in a SUFFIX=EDA synonym.
- Cross-platform joins.

For more information on passthru functionality, see the *TIBCO WebFOCUS® Adapter Administration* manual.

Avoiding Non-Automatic Passthru

If non-APT processing is critical to your extraction process, but is determined to be a bottleneck, it may help to add a secondary server to your configuration. The critical definition for Data Migrator (and its need to use a HOLD file) is the definition of APT at the server where Data Migrator is installed. For example, if you are reading from a C-ISAM file on UNIX (non-relational), describe the file to a second server configured expressly to avoid a non-APT condition. Server-to-server shipment of rows may outperform the disk I/O resulting from a HOLD file. This is illustrated in the following figure:



This technique requires that the additional server have access to a synonym describing the C-ISAM structure, and that the primary server is configured correctly for communicating with the secondary server. A synonym must be created to point to the correct destination. Data Migrator will use this new synonym for the SQL request, instead of the original C-ISAM Master definition (see the server manual for your platform for more information on configuring a hub and remote servers). Once again, the configuration work needed to avoid non-APT may be over-engineering in a situation where row volume, disk I/O, or processing power is not an issue.

Another way to avoid non-APT processing is to place your transformations in the synonym of the remote server. This is accomplished by entering a DEFINE in the synonym to ensure that calculating and processing work is done directly on the platform where the data resides.

Reference: Determining the Use of Automatic Passthru

You can tell if a flow is processed using APT or non-APT by looking at the Data Migrator ETL Log. A message appears towards the top of the output for every flow, indicating its capacity for passthru processing.

Reference: Using Real Numbers With Direct Passthru

Real or floating-point numbers, also referred to as approximate numeric, are numbers stored in scientific notation.

When numeric data is retrieved¿, it is converted to an intermediate display format. For fixed format files, FOCUS/FDS files, and relational data sources, the display format is that found in the synonym. However, when direct passthru is used, for example in a DBMS SQL flow, then Data Migrator does *not* use the synonym. In this case, Data Migrator uses a default format of 20.2 (that is, 20 digits with two places to the right of the decimal). This format may not be appropriate for data warehousing needs.

You can change the format with an SQL command. You can include this command in the profile on the remote server (where the data source resides). If you do so, all real numbers will be rounded to four digits.

If different tables in different data sources have different precisions, set the precisions for each process flow with a pre-extract procedure. If the data source resides at the server, create a pre-extract procedure with this command.

Syntax: How to Change the Format of Real Numbers for Direct Passthru

```
ENGINE database SET CONVERSION {FLOAT|REAL} PRECISION nn [mm]
```

where:

database

Is the type of data source to which you are connected.

nn

Is the precision. This number must be greater than 1 and less than 20.

mm

Is the scale (the number of places to the right of the decimal).

Example: Changing the Format of Real Numbers

For example, if your data source is Db2, and you want to set the number of digits to the right of the decimal to 4, issue the following command:

```
ENGINE DB2 SET CONVERSION FLOAT PRECISION 20 4
```

Reference: Direct Load Flows

Normally for fixed format file sources, Data Migrator extracts data from the files and creates an intermediate fixed format file. Then it invokes the load program to load from the fixed format file to target.

However, when the source for a Data Migrator flow is a single file, it is possible to load directly from that file using what is called a Direct Load Flow. This type of flow has no SQL object (there are no filters, sorts, or aggregations). There is a direct connection from source to target. Since there is no intermediate file created, this reduces disk space requirements, and can also reduce the time for the load.

Note that you cannot reference any DEFINEd columns in the synonym for the source. Also, each column in the source synonym must have an ALIAS.

The target table must be an existing table, and you can use Target Transformations.

Improving Data Loading Performance

You can improve performance when loading data into the data target. Keep the following suggestions in mind.

- **Prior to load options.** Before you load data into an existing table (for most RDBMS targets), Data Migrator gives you the option to Delete all rows from table or Truncate table. When deleting the rows, the relational database internally logs the rows deleted in case a ROLLBACK is issued, which for large tables can be slow. Truncating the table does not do any logging, so it is much quicker, but then a ROLLBACK cannot be issued.
- **Commit.** For relational database and FOCUS targets, Data Migrator lets you specify the number of rows before a COMMIT (or CHECK) is issued. The default value is 1000. A higher value generally improves throughput, but this depends on the memory available on the server, the database, the number of columns loaded, and their sizes. Increasing the commit value too much can slow performance. Try different values to see what works best.
- **Insert/Update.** This is the default load method. For a relational database target:

When include duplicates is selected each incoming row is sent to the target table without first checking to see if the key value is found. This generally provides the best loading time. If there is a duplicate key, the database will reject the row with a unique constraint violation. When reject or update duplicates is selected, the keys of each incoming row are matched against the target table. This additional step does slow processing, but allows for matched rows in the target table to be updated. In this case, care should be taken to use as few columns as keys as possible to uniquely identify the row. Note that if no columns are identified as key columns, the entire column is considered to be the key, which slows processing.

- **Using insert records from memory.** This load option uses a database vendor bulk or array insert API when available. This enables inserting a block of rows at a time. The default block size is 1000 rows, but a higher or lower value may provide better throughput, depending on the memory available on the server, the database, the number of columns loaded and their sizes. Try different values to see what works best.

  Note that incoming rows are not matched against the target table as with insert/update logic, so the number of rows inserted or rejected are not recorded. There are no flow statistics available.

  This method loads faster than the default load logic. For most databases, this is the fastest method, when the query can use automatic passthrough (so no intermediate flat file is created).

- **Using Bulk Load Utility using a disk file.** This load option uses a database vendor bulk load program. As the name implies, this method always creates an intermediate flat file. It is still faster than the default load logic. The options available depend on the database. For most databases, it is not quite as fast as Insert Records from Memory, when that option is available. However, for some databases designed for data warehousing (such as Sybase IQ), this is by far the fastest option. Additional tuning options are available depending on the database.

Improving Lookup Performance

The function DB_LOOKUP provides an alternative to a SQL JOIN to look up a value in one (or more) columns of a table. When matching values are found, the value of another column is returned.

For example, a lookup table may contain a list of state codes and names. If you look up the state code NY, you get back the state name New York.

This process can be expedited. By loading the pairs of values into memory (called a cache) you can reduce the number of file or database read operations.

There are two types of cache, depending on the type of the lookup table.

- **Static.** For a fixed format file, all the lookup values are loaded into the cache.
- **Dynamic.** For a relational database table, the lookup values and results are added into the cache as they are used.

If each input value is only looked up once, there is no advantage to using a cache. If the same input values are looked up multiple times, throughput can be improved because the return value is read from the cache instead of from the database.

For database lookup tables, a system setting controls whether the values in the lookup table are loaded into the cache.

Syntax: How to Enable the Lookup Cache

```
ENGINE INT CACHE {SET {ON [LIMIT  nn] | OFF } | STAT}
```

where:

`ON`

Enables caching.

`LIMIT` *nn*

Limits the number of rows added from cache to *nn*.

`OFF`

Disables caching.

`STAT`

Displays statistics for cache.

Example: Using the Lookup Cache

To enable the lookup cache for a data flow, create a stored procedure with the following line:

```
ENGINE INT CACHE SET ON
```

From the Process Flow tab for the flow, drag the stored procedure into the workspace. Connect the Start object to the stored procedure, and connect that to the data flow.

To see how the cache was used, create another stored procedure with the following line:

```
ENGINE INT CACHE STAT
```

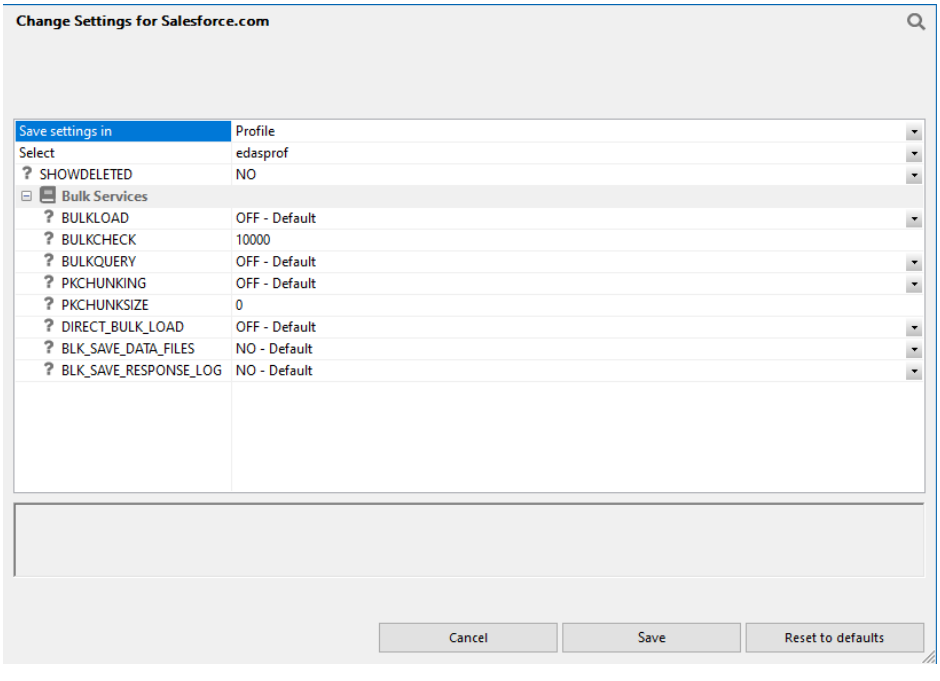
Drag it onto the process flow after the Data Flow. When you submit the flow, you will see lines, (like those shown in the following image), confirming that the cache was used.

```
(FOC1300) : CACHE ID = 'DMINV PROD_NUM#PRODNAME'
(FOC1300) :     Stored fields = PRODNAME
(FOC1300) :     Type = DYNAMIC, Records = 16
(FOC1300) : CACHE ID = 'DMCOMP STORE_CODE#STORENAME'
(FOC1300) :     Stored fields = STORENAME
(FOC1300) :     Type = DYNAMIC, Records = 138
```

Bulk Retrieval With Salesforce.com

When working with Salesforce.com, there are various settings that you can set that will allow you to query large data sets and reduce the number of API requests.

These settings can be accessed from the Change Settings dialog box for Salesforce.com. To open this dialog box, from the DMC, expand the *Adapters* folder, right-click *Salesforce.com*, and click *Change Settings*.



The following Bulk Services options are available:

BULKLOAD
Enables the use of Extended Bulk Load. The default value is OFF.

BULKCHECK
The row interval to commit or write transactions to the database. The default value is 10000. If the value is set to 0, the commit is issued when the job is completed.

BULKQUERY
Enables Bulk Query for data retrieval. The default value is OFF. When this option is selected, queries that meet the Salesforce.com requirements, such as no aggregation, will use bulk query.

You can also use the following command to enable or disable Bulk Query:

```
ENGINE SFDC SET BULKQUERY {ON|_OFF_}
```

PKCHUNKING
Enables bulk query Primary Key chunking for data retrieval. The default value is OFF. When this option is enabled, extremely large datasets can be extracted. The Primary Key of the object is used to retrieve data in chunks, which allows for retrieval of larger data values.

You can also use the following command to extract extremely large datasets:

```
ENGINE SFDC SET PKCHUNKING {ON|_OFF_}
```

PKCHUNKSIZE
Enables the bulk query Primary Key chunk size for data retrieval. The default value is 0. When this value is not set, or set to 0, Salesforce.com will use the default chunk size of 100000. The maximum chunk size is 250000.

You can also use the following command to enable the bulk query chunk size:

```
ENGINE SFDC SET PKCHUNKSIZE {0..250000}
```

BLK_SAVE_DATA_FILES
When set to *Yes*, retains an intermediate file that is created, as well as the log (response) file. The default value is *No*.

The files are saved in the same application directory as the flow. Each is generated with a synonym that you can use for reporting. The file type of the intermediate HTML file is .ftm and the response file is .log. You can use these files in conjunction with the batch results to see which records were loaded successfully (with the internal id for those records), and which were not (with the error message for those records).

BLK_SAVE_RESPONSE_LOG

When set to *Yes*, saves the response log file. The log file will automatically be saved if BLK_SAVE_DATA_FILES is set to *Yes*. The default value is *No*. The file is saved in the same application as the flow and has the file type .log.

Chapter 10

Working With Synonyms

The Data Management Console allows you to view, edit, and create synonyms. It includes a Synonym Editor, which provides a graphical interface for working with synonyms.

Editing synonyms can be particularly useful in Data Migrator for adding virtual columns that perform transformations.

The Synonym Editor also provides access to Data Profiling and Data Lineage information.

In this chapter:

- Synonym Editor Layout
- Viewing and Editing Synonym Attributes
- Using Variables in a Synonym
- Variables Attribute Summary
- Use a Variable in a Synonym
- Creating and Enhancing Synonyms Using Modeling View
- Adding Indexes to a Synonym
- Adding Virtual Columns (DEFINE) in a Synonym
- Viewing Data Profiling Characteristics
- Viewing Data Lineage for a Column

Synonym Editor Layout

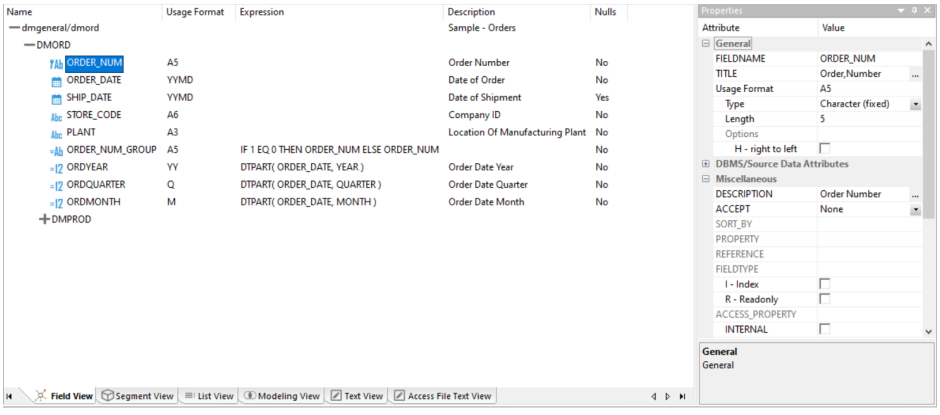
The Synonym Editor occupies the DMC workspace and displays the Field View, Segment View, Modeling View, Text View, and Access File Text View tabs.

Reference: **Synonym Editor - Field View Tab**

The Field View tab shows a hierarchy of segments and columns on the left, with the attributes and values of the selected item on the right.

Note: The attributes available depend on the type of synonym.

The image below is an example of an SQL data source with a key column selected.



Note: Information about the attribute that has focus is displayed at the bottom of the attribute list. In this case, an explanation of the FIELDNAME attribute appears.

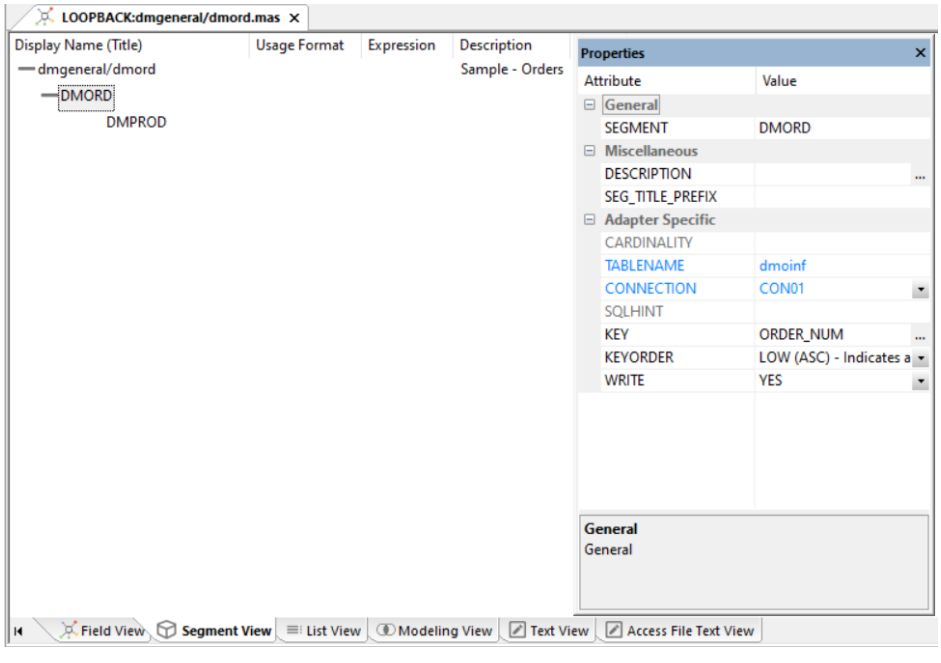
The following objects may appear on the Field View tab.

Object	Function	Description
—	Synonym	General icon used to indicate synonyms, visible on top of the Field View tab. It provides information about the file name used and the application it resides in.
—	Segment	Indicates a segment in a synonym. The root or parent segment appears first in the tree.
?Ab	Key Column	Indicates a key column. **Note:** The icon will vary based on the format of the column.
Abc 123	Column	Indicates a general column. **Note:** The icon will vary based on the format of the column.
=12	Virtual Column (DEFINE)	Indicates a virtual or defined column. **Note:** The icon will vary based on the format of the column.

Object	Function	Description
	Index Column	For FOCUS files, indicates the field index for quick retrieval of values for this field.
	Filter	Indicates a filter in the synonym.
	Compute	Indicates a computed field.
	Internal field	Indicates a column that does not appear in sample data or in the list of available columns.
	Needs Value field	Indicates a column that requires a value to access the data.

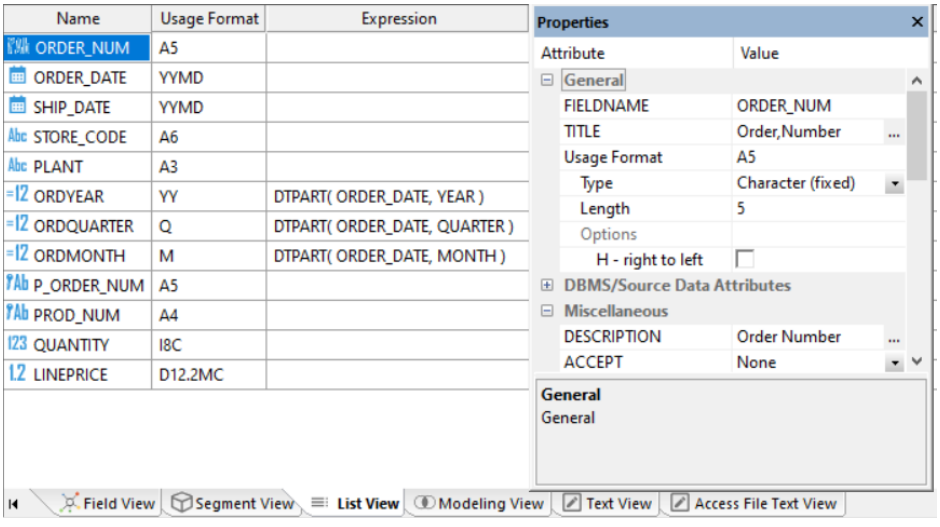
Reference: **Synonym Editor - Segment View Tab**

The Segment View tab shows a hierarchy of segments on the left, with the attributes and values of the selected item on the right, as seen in the following image.



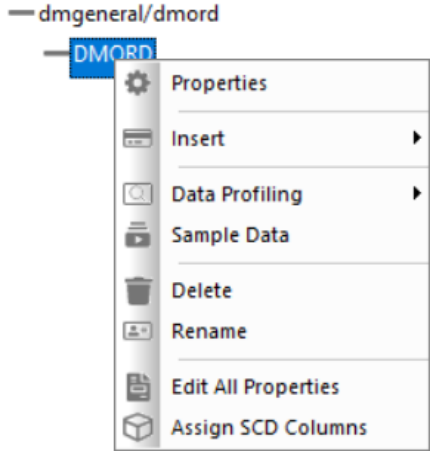
Reference: Synonym Editor - List View Tab

The List View tab shows a list of objects on the left, with the attributes and values of the selected item on the right, as shown in the following image.



Reference: Synonym Editor - Segment Shortcut Menu

When you right-click a segment in the Synonym Editor, the shortcut menu shown in the following image appears in both the Field View and Segment View tabs. This image shows a SQL data source with the root segment selected.



The following options are available:

Properties

Opens a pane showing the properties of the selected segment.

Insert

Enables you to insert one of the following:

Reference to Existing Synonym as Child

The reference is a pointer to the source synonym using the synonym as a child. If you subsequently make changes to the source synonym, reopening the current synonym will reflect those changes.

Reference to Existing Synonym as Root

The reference is a pointer to the source synonym using the synonym as a root. If you subsequently make changes to the source synonym, reopening the current synonym will reflect those changes.

Segment via Metadata Import

Enables you to create and add a new synonym to the current synonym using the Create Synonym tool.

Segment Manually

Adds a synonym that must be coded manually.

Field

Adds a general column to the segment.

Define

Adds a virtual or defined column to the segment.

Filter

Adds a filter to the segment.

Compute

Adds a calculated value to the file.

Sort Object

Adds a Sort Objects folder and a sort object. This option is available only if the Support extended options option is selected and is not used with Data Migrator.

Variable

Adds a Variables folder and a variable to the segment.

Style

Adds a Styles folder and a style object. This option is available only if the Support extended options option is selected and is not used with Data Migrator.

Subquery

Adds a subquery to the fields list.

Index

Adds an Index folder and an index.

Data Profiling

Provides the characteristics of the data for a segment. The following options are available:

Statistics

Shows statistic values for the selected segment only.

Count

Shows a count of the number of rows in the selected source table. For a multi-segment synonym, it shows the count for each segment.

Statistics with parent segment

This option is available only when viewing a lower segment of a hierarchy (cluster join).

Key Analysis

Allows you to select the columns that you would like to analyze as potential keys to the source table.

Values

Shows unique values and their percentages.

Duplicate Values

Shows identical values and their percentages.

Sample Data

Displays sample data in the workspace for the selected segment or synonym.

Sample Data with parent key

Displays sample data for the selected segment joined to the parent segment. This option is only available when a child segment is selected in a multi-segment synonym.

Delete

Deletes the segment.

Rename

Allows you to rename the segment.

Edit All Properties

Allows you to edit adapter specific properties, such as the delimiter value in a delimited file.

Assign SCD Columns

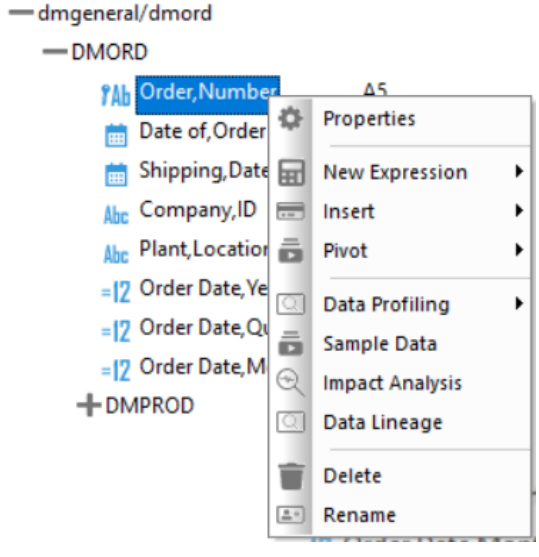
Allows you to assign columns to a SCD type by using drag-and-drop.

Join Properties

Opens the Join Editor so that you can specify how the selected segment is joined to the parent segment. This option is only available when a child segment is selected in a multi-segment synonym.

Reference: Synonym Editor - Column/Field Shortcut Menu

When you right-click a column in the Synonym Editor, a shortcut menu appears in both the Field View and List View tabs. The following image shows a SQL data source with a column selected.



Note: If you multi-select columns and right-click them, an abridged version of the shortcut menu opens.

The following options are available:

Properties

Opens a pane showing the properties of the selected column.

New Expression

Displays a shortcut menu with a list of function types that expands to show a list of available function that can be added to the synonym. Selecting one of the functions opens the Function Assist dialog box, which allows you to specify parameters for that function. The functions shown will vary and are dependent on the data type originally selected.

Insert

Enables you to insert one of the following:

Field

Adds a general column to the synonym.

Date/Time Function

Inserts a Date/Time function.

Filter

Adds a filter to the segment.

Grouping

Opens the Group values for dialog box, where you can configure group values.

Aggregation Function

Provides a choice of preconfigured aggregation functions that open in the Function Assist dialog box. The selected functions will be saved in the Constant Defines/Computes folder for the synonym.

Analytic Function

Provides a choice of preconfigured analytic functions that open in the Function Assist dialog box. The selected functions will be saved in the Constant Defines/Computes folder for the synonym.

Sort Object

Adds a Sort Objects folder and a sort object.

Variable

Adds a Variables folder, if needed, and a variable.

Style

Adds a Styles folder and a style object.

Subquery

Adds a subquery to the fields list.

Pivot

The following available:

Multiple columns to rows

Allows you to turn, or pivot, columns into rows or a single column into multiple columns, for retrieval.

Impact Analysis

Displays an Impact Analysis report for the particular column in the workspace. An Impact Analysis report identifies the procedures that access a Master File or field within a Master File.

Data Lineage

Provides the ability to see how the column was derived.

Data Profiling

Provides the characteristics of the data for a column. For more information, see *Viewing Data Profiling Characteristics* on page 530.

Statistics

Shows statistic values for the selected segment only.

Key Analysis

Allows you to select the columns that you would like to analyze as potential keys to the source table.

Hex View

Shows hexadecimal value of the selected field.

Values

Shows a values report for the selected column.

Values Graph

Displays a graph for both alpha and numeric field types.

Values Pie Graph

Displays a pie graph.

Duplicate Values

Shows identical values and their percentages.

Outliers

Shows outlier report for the selected column.

Sample Data

Displays sample data in the workspace.

Delete

Deletes the column.

Rename

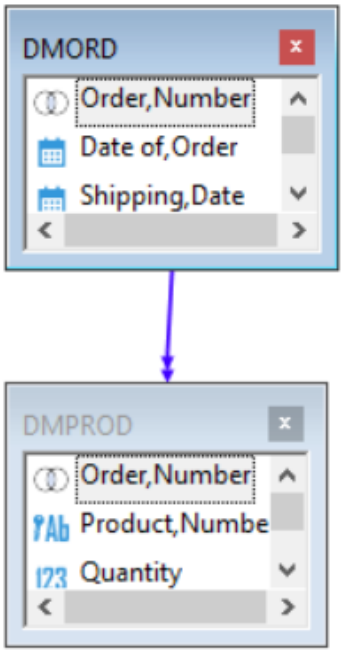
Allows you to rename the column.

Decompose Date

For a date or date-time field, opens a dialog box for selecting the date components that should be defined as virtual fields.

Reference: **Synonym Editor - Modeling View Tab**

The Modeling View tab shows a graphical representation of the synonym, as shown in the following image. Use the Modeling View to define dimensions for OLAP analysis, view join properties, create cluster joins, and add or edit segments.



Reference: Synonym Editor - Text View Tab

The Text View tab shows the description of the underlying table.

You can print the synonym syntax by selecting *Print* from the Application menu.

The image below is an example of an MS SQL Server data source in the Text View tab.

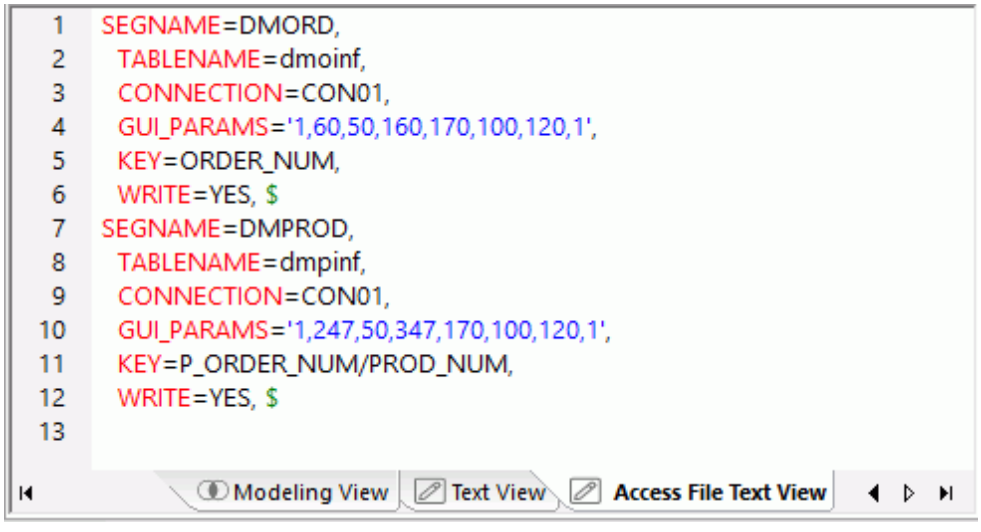


Reference: Synonym Editor - Access File Text View Tab

The Access File Text View tab shows the description of the Access File for a synonym, which is used to access the database.

Note: You can print the synonym syntax by selecting *Print* from the Application menu.

The image below is an example of an SQL data source in the Access File Text View tab.



Note: While you can edit the text views, it is highly recommended that you do so with care. Editing the synonym incorrectly can invalidate the synonym.

Reference: Metadata Tab

The Metadata tab in the ribbon contains groups and commands that provide quick access to commonly performed functions while working in the Synonym Editor. The behavior of the command is determined by the selected object. Therefore, certain tab commands may be inactive.

The Metadata tab contains the following additional groups:

View Group

Field

Switches to the Field View in the workspace.

Segment

Switches to the Segment View in the workspace.

List

Switches to the List View in the workspace.

Modeling

Switches to the Modeling View in the workspace.

Text

Switches to the Text View in the workspace.

Access File Text

Switches to the Access File Text View in the workspace.

Aerial View

Opens the Aerial View window. This view is only available when in the Modeling View tab.

Insert Group

Insert

The Insert command allows you to insert one of the following:

- Reference to Existing Synonym as Child
- Reference to Existing Synonym as Root
- Copy of Existing Synonym
- Segment via Metadata Import
- Segment Manually
- DBA
- Field
- Define
- Filter
- Compute
- Group
- Sort Object
- Variable

- Style

Reports Group

Sample Data

Enables you to view and refresh sample data for the object.

Data Profiling

Provides the data characteristics for synonym columns. The following options are available:

- Statistics
- Count
- Key Analysis

Impact Analysis

Runs the impact analysis report for the selected column.

Tools Group

Properties

Displays the Properties panel for the selected object.

Quick Copy

This option is only available when the synonym name (root) is selected. For more information, see *Creating a Quick Copy* on page 367.

DBA

Adds field/value security DBA.

Business View

Enables you to create a Business View and a custom Master File that can use selected fields from the original synonym. In addition, you can customize field names, titles, and descriptions.

Viewing and Editing Synonym Attributes

The Synonym Editor enables you to view and edit the attributes of a synonym.

Procedure: How to View and Edit Synonym Attributes

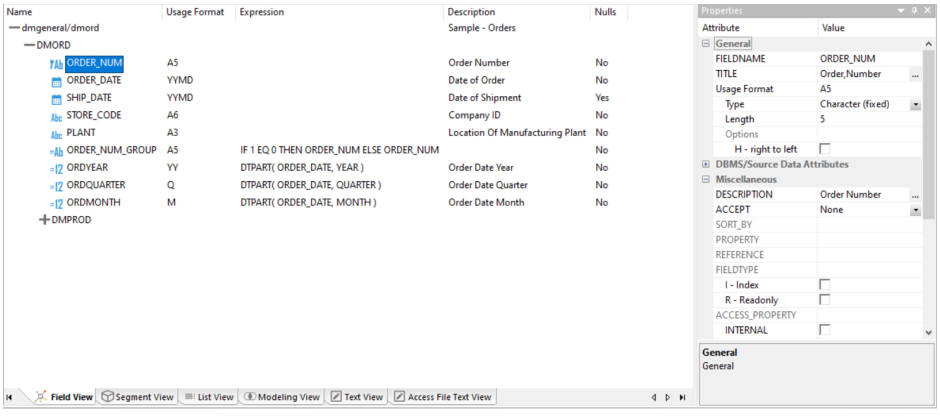
To view and edit synonym attributes:

1. Double-click a synonym, or right-click it and click *Open*.

The Synonym Editor opens to the Field View tab, which shows a hierarchy of segments and columns on the left, with the attributes and values of the selected item on the right.

The attributes available depend on the type of synonym.

The image below is an example of an SQL data source with a key column selected.



The values for Usage Format, Expression, Description, and Nulls are viewable in the hierarchy with the columns. To edit these values, use the corresponding attribute fields in the Properties pane on the right-hand side of the Synonym Editor.

2. You can change the attribute values by typing in new values or by using the drop-down menus and check boxes.

   The Synonym Editor does not let you make any changes that would render the synonym unusable. Therefore, you cannot edit any value field that is highlighted gray. In addition, if a change does not have proper syntax or format applied, the field may appear in red text. Messages and warnings appear if you try to save a file that contains an error.

3. Save changes by clicking *Save* from the Quick Access Toolbar.

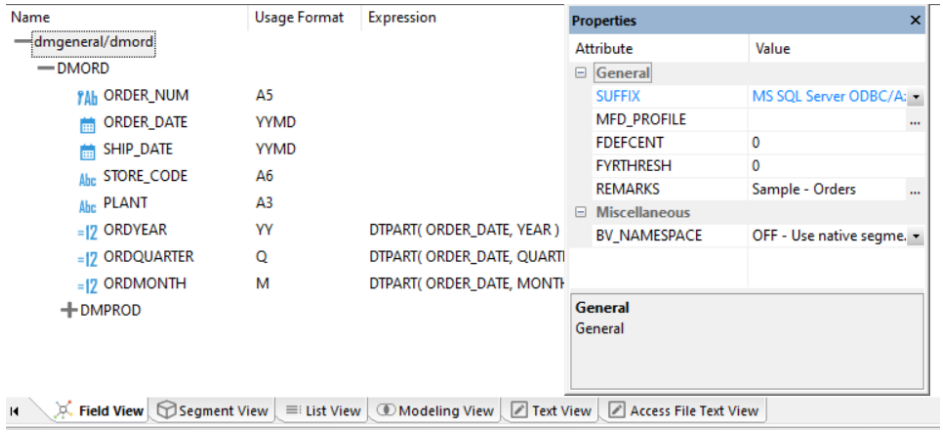
   Note: You can multi-select fields to change attributes for multiple fields in a single operation. The display changes to show only those attributes that can be changed.

4. Close the Synonym Editor by clicking the X.

Note: If you close the Synonym Editor without saving your changes, you are prompted to do so.

Reference: File Attributes Summary

The image below is an example of an SQL data source with the synonym file name selected.



Note:

- Information about the attribute that has focus is displayed at the bottom of the attribute list.
- The attributes available depend on the type of synonym.

Synonyms can have the following file attributes:

General

SUFFIX

Identifies the type of synonym or data source.

MFD_PROFILE

The name of the foxexec that will be executed before the request containing MFD.

FDEFCENT

Defines the default century value, specifying a century number for handling cross-century dates.

Note: Use the default setting (0) unless you wish to retrieve data from a earlier century (for example, 19xx).

FYRTHRESH

Defines the base years, to represent the lowest year to which the century value applies (FDEFCENT).

Note: Use the default setting (0) unless you wish to retrieve data from a earlier century (for example, 19xx).

REMARKS

Enables you to include descriptive information at the file level and specify multiple language descriptions for the synonym. Remarks are displayed along with the file name during reporting.

DATASET

Identifies the location of the data source to be used in the file name, including the extension and the location of the data file. This option is only available for file data sources, such as fixed format files or XML and JSON documents.

BV_NAMESPACE

Controls the Business View run-time resolution strategy.

DV_INCLUDE

Controls the base file DB info inheritance in a cluster.

Reference: Segment Attributes Summary

If a synonym segment is selected, the attributes listed below are available.

Note:

- Information about the attribute that has focus is displayed at the bottom of the attribute list.
- The attributes available depend on the type of synonym.

General

SEGMENT

Is the name of the segment.

SEGTYPE

Specifies the type of relationship that a segment has to its parent, and indicates which of the segment fields are key fields and in what order they are sorted.

Type. Identify the segment type and sorting options from the Type drop-down list.

KEY. Enter the names of key fields that you want to use for sorting. For example, no two employees can have the same employee ID number, so you can use that field as the key. A segment instance can have more than one field that makes up the key. That is, two or more field values may be used to distinguish records.

SEGSUF

SEGSUF is used when part of the data source being described by the synonym is of a different data source type than that declared for the entire structure. This option is not available for the first segment in a synonym.

Note: SEGSUF is the data source type of a segment and any descendants it might have, where that type differs from the SUFFIX value.

Miscellaneous

DESCRIPTION

Contains a description or comments about the segment.

CRFILENAME

Is the name of the cross-referenced data source.

CRSEGNAME

Is the name of the cross-referenced segment.

CRKEY

Identifies the common join field for the cross-referenced segment.

Note: These cross-referenced values (CRFilename, CRSegname, CRKey) are available for FOCUS data sources.

SEG_TITLE_PREFIX

An optional attribute that enables you to provide the title prefix for the segment fields that appear in reports. You can split the text across up to five separate title lines by separating the lines with a comma.

Adapter Specific

Adapter specific fields are shown if an Access File component has been generated with the synonym.

CARDINALITY

Defines how many members of a dimension can be retrieved for a report.

TABLENAME

Identifies the table or view. It may contain the owner ID, as well as the table name. For some synonyms, it must also contain the data source name. This value may be a variable.

CONNECTION

Indicates the host server or data source for synonyms. This value may be a variable.

KEY

Identifies the names of the columns that constitute the primary key.

KEYORDER

Identifies the logical sort sequence of data by the primary key.

WRITE

Specifies whether write operations are allowed against the table.

DBSPACE

Identifies the storage area in which the table resides.

PERSISTENCE

Specifies the type of table persistence and related table properties. This is optional for database management systems that support volatile tables, and required otherwise.

Reference: Column/Field Attribute Summary

If a column or field in a synonym is selected, the attributes listed below are available.

Note:

- Information about the attribute that has focus is displayed at the bottom of the attribute list.
- The attributes available depend on the type of synonym.

General

FIELDNAME

Is the name of the column.

TITLE

Supplies a title to replace the column name that is normally used in reports and enables you to specify multiple language titles for the column or field.

Usage Format

Describes the data type and format for the column for usage or display.

Note: Additional attributes, DEFCENT and YRTHRESH, are available if the Usage field is set to Date, Time, or DateTime (Timestamp) format. Use these attributes to enter the century and year threshold values for the column or field.

ALIAS

Assigns an alternative name for a column, or the real column name for a DBMS synonym.

For file data sources, a special reserved ALIAS of INSTANCE means the associated FIELD will show the name of the file when it is read.

When a FIELDNAME of RECTYPE is used, the ALIAS contains the value that identifies the record type.

Key Component

Indicates whether the field is part of the primary key for the table.

ACTUAL

Describes the type and length of data as it is actually stored in the data source.

MISSING

Controls how null data is handled, that is, if no transaction value is supplied. This option is not available for KEY fields.

Miscellaneous

DESCRIPTION

Contains a description or comments about the column or field.

ACCEPT

Specifies criteria for validating data.

- **OR** enables you to specify an acceptable value.
- **SYNONYM** enables you to specify a synonym as the source of the accept list.
- **FOCEXEC** enables you to specify a FOCEXEC as the source of the accept list.
- **FROM-TO** enables you to specify a range of acceptable value fields.
- **FIND** enables you to supply file and field names to instruct where to search for a data source and for a list of acceptable values. You supply the field name of the data field for which the validation criteria are being assigned, the file name of the target FOCUS data source where the field can be found, and the field name of the target data field that contains the validation criteria.

FIND is only available for FOCUS data sources and does not apply to OLAP-enabled synonyms. Also note that, in the Maintain environment, FIND is not supported when developing a synonym.

WITHIN

Contains the name of a field to be included in a dimension.

PROPERTY

Describes which hierarchy property the column represents.

REFERENCE

Points to the field that is the unique identifier for the hierarchy.

FIELDTYPE

- **I - Index** identifies an indexed column in a FOCUS database.
- **R - Read only** indicates a read-only column. This setting is useful for columns that are automatically assigned a value by the RDBMS.

ACCESS_PROPERTY

Specifies access options for the column data.

- **INTERNAL** defines a column that does not appear in sample data or in the list of available columns. This restricts the field from showing in any of the field lists in the reporting tools.
- **NEED_VALUE** defines a column that requires a value to access the data.
- **Select By** defines a column by value, range, or multi-values.

HELPMESSAGE

Appends a help message to a column.

This attribute is not used by WebFOCUS.

SCD Type

Sets slowly changing dimension attributes. This option is only available for existing relational targets and is only used by a flow when SCD is enabled in the target object.

A surrogate key is the KEY column in the table and has an SCD type of blank. Other columns with a blank SCD type have no SCD processing done to them. In a synonym, this column will always appear first, even if it is not the first column in the table.

Note: The surrogate key must be an integer. If the column is identified in the synonym as read-only and uses AUTOINCREMENT, then the database native processing is used to assign surrogate key values. If it is not so identified, then Data Migrator processing assigns values for this column, incrementing for each new row added.

- **Logical Key Field.** Is the source database key. Multiple columns may comprise the key.

  The best practice, for faster run-time performance, is to ensure that a non-unique index is set for the Logical Key field in the SCD synonym.

- **Begin Date/End Date for Type II changes.** Indicates date range for the row values. A null end date indicates the row is current. If you use Begin and End Date, the End Date column must be created as nullable, unless you override the default End Date value of NULL with a value for &&CM__SCDAEDATE.

  The data type must be either a Date format or a Date and Time format. However, the formats of the two fields must be identical.

  Begin/End Date values for Slowly Changing Dimensions can also be Datetime or Timestamp values.

- **Change Flag.** For Type I changes, a value is assigned when a row is updated. If a value is not assigned in Target Transformations for the Date and Datetime columns, the current date or timestamp is assigned automatically.

- **Activation Flag.** Indicates if the row is current.

  This field must be an integer or a character A20V. Valid values when this field is described as an integer are:

  - **1**. Indicates that a record is active.
  - **0**. Indicates that a record is inactive.

  When this field described as a character field, you can still use the values 1 and 0, or you can assign other values. To assign values, you can use the Set Variables object on a Process Flow tab and set values for the global variables CM_SCDACT and CM_SCDINACT.

- **Type I** (overwriting history) designates columns whose database values are overwritten with new values.
- **Type II** (preserving history) designates columns whose database rows are flagged as inactive or assigned an end date. New rows are inserted with the new values.
- **blank** (non-key permanent columns) indicates that database values are not changed.

TEMPORAL_PROPERTY

Inserts virtual fields in the Master File to show how selected totals progressed through a time period for specified intervals.

USE_STYLE

Are the styles (for example, font and color) to apply to the field.

The attributes available depend on the type of synonym.

Adapter Specific

AUTOINCREMENT

Indicates that the field is automatically incremented by the relational database when rows are inserted. This option is only available for selected relational databases. When selected, the field must also be identified with a field type of read-only.

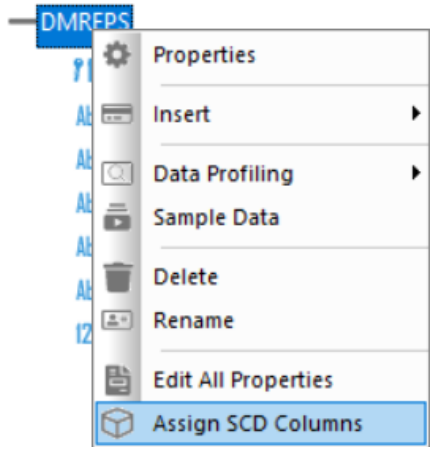
The following attributes are only used if AUTOINCREMENT is checked.

- **START.** Indicates the starting value for this AUTOINCREMENT field. If not specified, 1 is used.
- **INCREMENT.** Indicates the increment value for this field. If not specified, 1 is used.
- **SEQUENCE.** Is the name of the corresponding database SEQUENCE object for this field. This option is only available for ORACLE tables.

Procedure: How to Assign SCD Columns on a Synonym

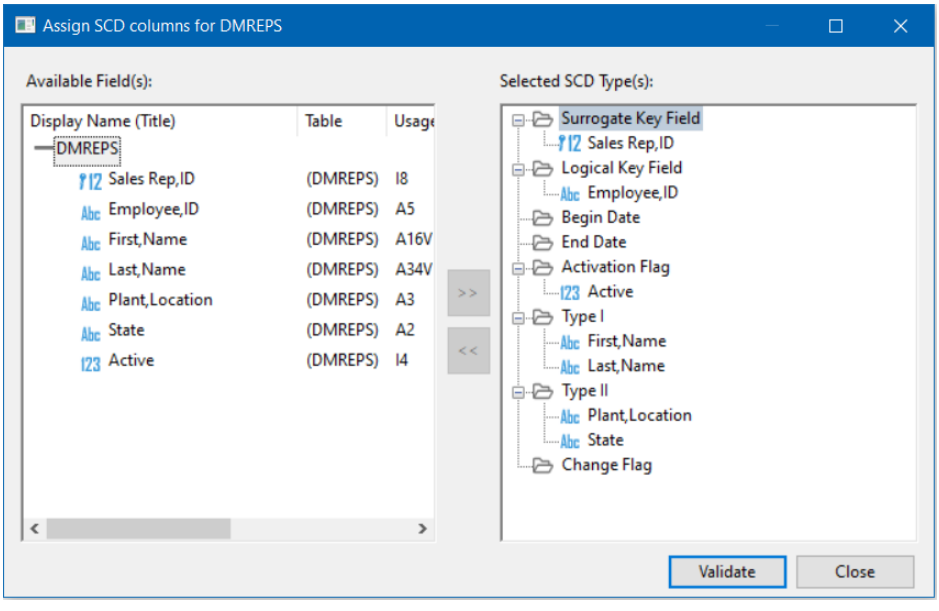
You can assign SCD (Slowly Changing Dimension) Types column attributes from a single dialog box in the Synonym Editor. This operation is performed on the segment level from the Field View, Segment View, or Modeling View.

1. In the Synonym Editor, right-click the segment name and click *Assign SCD Columns* from the shortcut menu, as shown in the following image.



The Assign SCD columns dialog box opens.

2. To assign the SCD Type to a column, click the column in the Available Field(s) section and the SCD type in the Selected SCD Type(s) section, followed by the right arrow. You can also drag the column from the left to the desired SCD Type on the right. An example of assigning SCD types to a column is shown in the following image.



3. Click *Validate* to confirm that all SCD Types are assigned to the synonym columns correctly.

   If the validation passes, the synonym can be used as a target synonym in a Data Flow with the Load Type Property set to *Slowly Changing Dimensions.*

   If the validation does not pass, the segment name appears in red. To view the explanation of the error, right-click the segment name and click *Error Details*.

Pivoting Repeating Columns in a Synonym

When you have multiple columns or groups of columns that represent identical values, you can pivot the column or group of columns so that each value becomes its own row.

- **Repeating column.** This transformation pivots each row with a consecutive set of semantically identical data columns into rows. For report generation, three new columns are created.

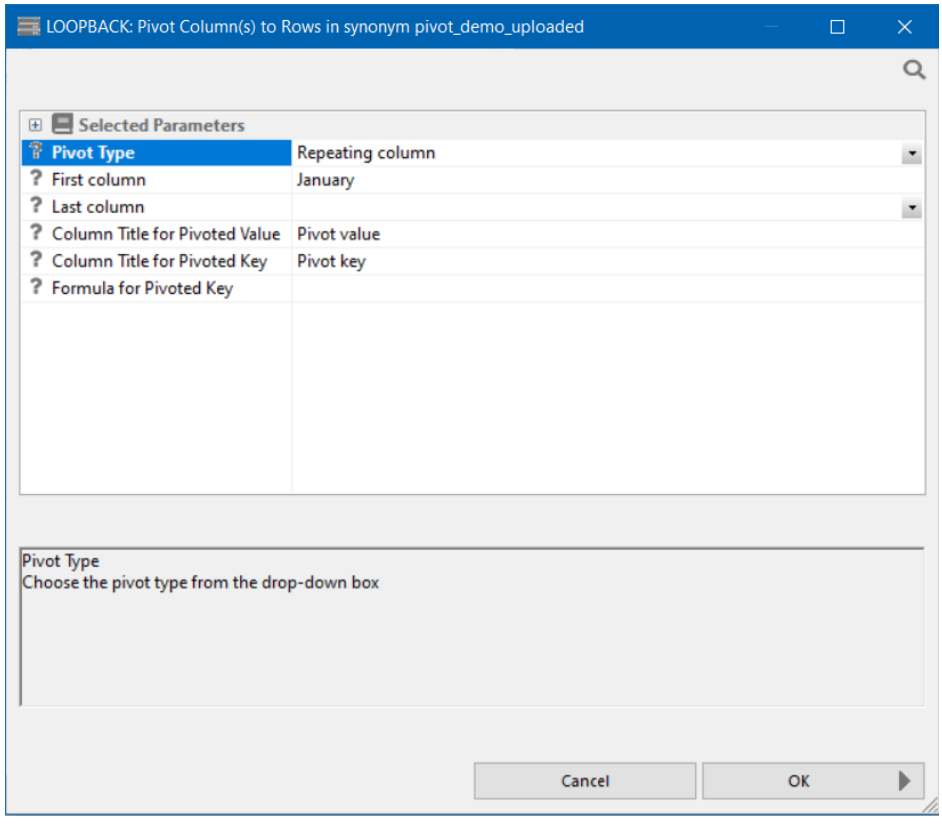
- **Repeating group of columns.** This transformation pivots each row with a consecutive set of semantically identical groups of data columns each into rows. For report generation, new columns are created for each column in the original group plus one additional column to identify the grouping.

In each case, fields are created with the information needed to order and display the rows. For numeric values. You can edit the formulas used to do this, but it is not recommended when the field headings are numeric.

Procedure: How to Pivot Repeating Columns in a Synonym

1. Right-click the first column that you want to transpose, point to *Pivot*, and click *Multiple columns to rows*.

   The Pivot Columns to Rows dialog box opens, as shown in the following image.



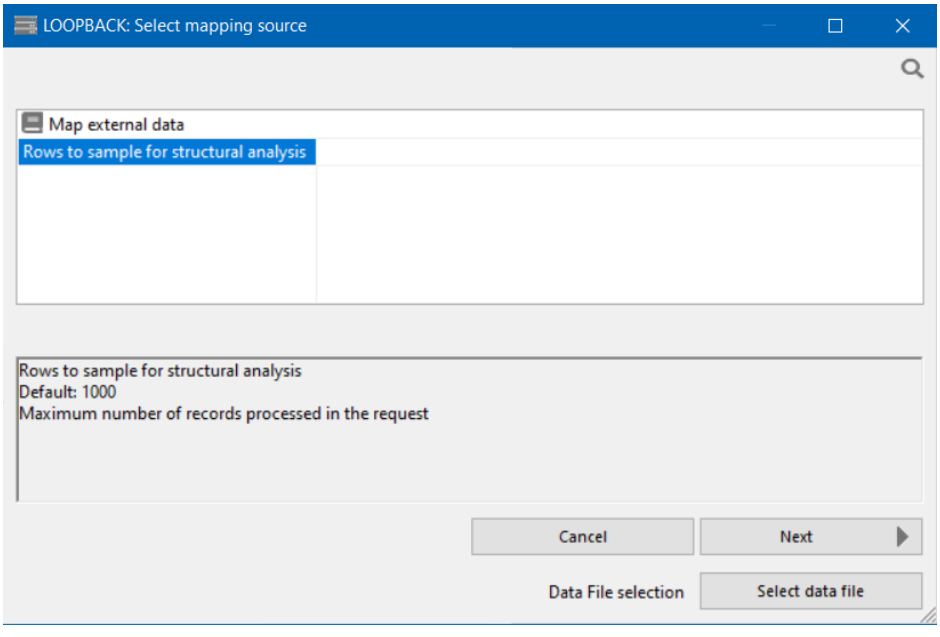
2. Select the last column to pivot.

3. Click *OK*.

Procedure: How to Pivot Repeating Groups of Columns in a Synonym

1. Right-click the column that you want to transpose, point to *Pivot*, and click *Multiple values to rows*.

   The Select mapping source dialog box opens, as shown in the following image for groups.



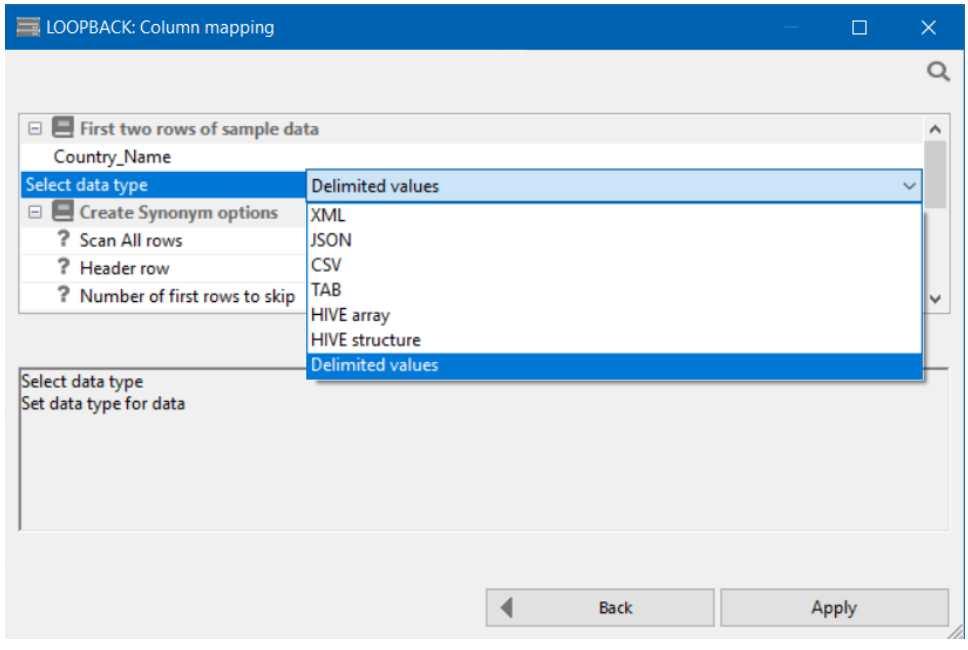
   The Rows to sample for structural analysis box allows you to specify the number of rows to examine to determine the structure and data types of the values in the column. Enter a value here, or leave it blank to use the default value of 1000.

   If the source column is empty, or contains unrepresentative data, you can choose to use an external file as a prototype instead of the column values. If the source column contains XML and you have an XSD (XML Schema Definition) that describes the structure, you can use it. For either option, click *Select data file*.

2. Click *Next*.

   The Column mapping dialog box opens

3. Select a data type from the drop-down menu, as shown in the following image.



The drop-down menu shows you the suggested data type by default. To help you identify the type of data that is stored in the selected column, the first two rows of data are shown above the drop-down menu.

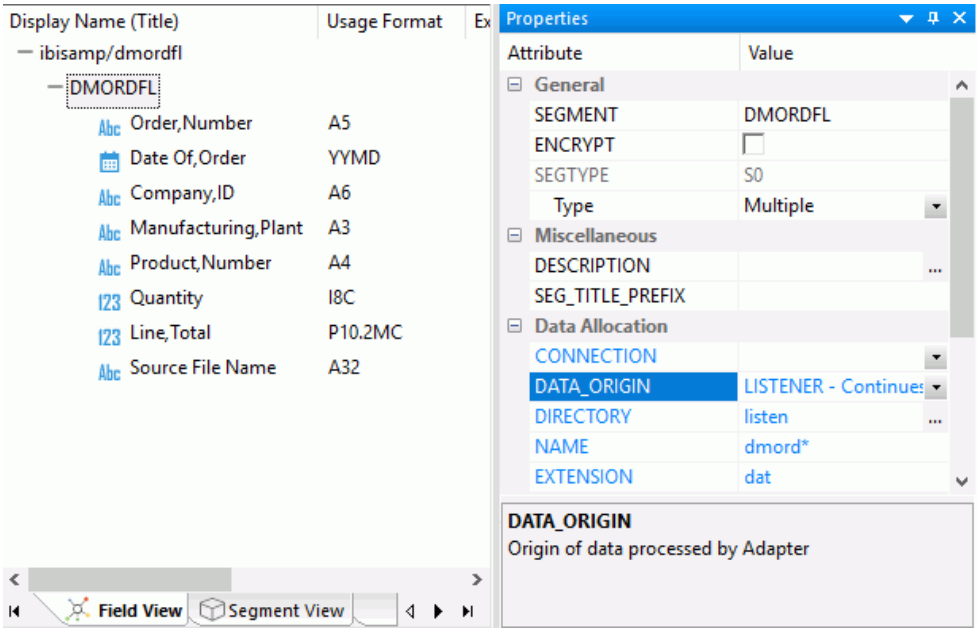
The available data types are:

- Delimited values. You must supply additional information, such as delimiter.
- XML (eXtended Markup Language)
- JSON (JavaScript Object Notation)
- CSV (Comma Separated Values)
- TAB (Tab Separated Values)
- Hive Array (List of values. Each one becomes a separate row.)
- Hive Structure
- Delimited Values (specify delimiter and optional enclosure)

4. Click *Apply*.

Reference: File Listener Attributes

The image below is an example of a fixed format file data source with DATA_ORIGIN of LISTENER.



A fixed format file data source may have the following attributes:

Data Allocation

CONNECTION

The name of a connection for a ffixed formatlat file source that identifies an FTP or SFTP server. This field is optional. If not specified, the file resides locally.

DATA_ORIGIN

Indicates the origin of data described by the synonym.

FILE. Indicates that a file (or collection of files if a wildcard is used in the dataset parameter) is read once. This is the default.

LISTENER. Indicates that files are read as they appear, as specified in the remainder of the File Collection Parameters.

TAIL. Is reserved for future use.

DIRECTORY

Defines the location of data files for polling. This can be an application directory, a mapped application name, or a directory location on your file system.

When using a directory on a file system for local files, the name of the physical directory can be used.

When using a directory on a file system for remote data files, a directory relative to the initial directory can be used. For example, if your remote initial directory is /home/user1/apps, using *sales* will set the data files directory location to /home/user1/apps/sales.

NAME

Indicates the name of the file. May include wildcard characters. The default is *.

EXTENSION

Indicates the file type that the listener is checking. The default is .ftm.

The parameters below only apply to Origin of Listener.

Pre Processing Actions

PICKUP

Indicates the mechanism used to pick up files from their directories. The options are:

IMMEDIATE. Processes the file as soon as it is detected. This is the default.

TRIGGER. Processes the file when a trigger file is detected. This setting requires an additional setting for the trigger file extension.

A trigger is typically used when processing needs to be delayed. For example, when using FTP to copy files, if you wait until this secondary file appears while a large file is being copied into a directory, the trigger file indicates when processing can begin.

When a pickup strategy of trigger is used, indicates the trigger file extension. The trigger file extension is added to the full name of the file being listened for. For example, if the trigger file extension is .trg, the trigger file for a data file named input.dat would be input.dat.trg. There is no default value.

MONITOR. Monitors the file as soon as a file change is detected.

The files themselves are not read, only their name and location are recorded. This processes files and sub-directories when they are added, changed, or deleted.

Post Processing Actions

DISCARD

Indicates what to do with the file after it has been processed by a file agent. The options are:

DELETE. Deletes the file.

ARCHIVE. Moves the file, while adding a timestamp to the name, to a specified directory. This is the default.

ARCHIVE_LOCAL. Defines the location where the data files will be archived. This location can be an application directory, a mapped application name, or a directory on your local file system.

This option must be used only for a CONNECTION that is set with FTP or SFTP protocol.

This option allows archiving the files locally, where the Server is running, while the listening directory is on a remote location.

KEEP. Keeps the file. This option is only available for the Trigger pickup type.

ARCHIVE

Defines the location where the data files will be archived when DISCARD strategy of ARCHIVE is used. This can be an application directory, a mapped application name, or a directory location on your file system.

When using a directory on a file system for local files, the name of the physical directory can be used.

When using a directory on a file system for remote data files, a directory relative to the initial directory can be used. For example, if your remote initial directory is /home/user1/apps, using *sales* will set the data files directory location to /home/user1/apps/sales.

Listening Parameters

POLLING

Indicates the polling interval (in seconds) for new files arriving in the specified directory. For example, a polling interval of 10 means the directory is scanned every ten seconds for new files. The default is 10 seconds.

TIMEOUT

Indicates the timeout interval (in seconds). If there is no activity in this time interval, the processing will stop. A Zero value means there is no limit. The default value is 1 second.

MAXERRORS

The maximum number of errors allowed.

MAXQSIZE

The maximum number of files in the processing queue.

QSORT

The sort order for files in the processing queue.

Read Limits

MAXFILES

Indicates the maximum number of data files processed by the request. The default is 99999999.

MAXRECS

Used only for the data origin of TAIL. Indicates the maximum number of records processed by the request. The default is 99999999.

Adapter Specific

MISSVAL_CHAR

Indicates a missing alpha value in the data source.

MISSVAL_NUMB

Indicates a missing numeric value in the data source.

MISSVAL_DATE

Indicates a missing date/time/timestamp value in the data source.

Reference: File Listener for Use With Excel Sheets

The File Listener can be used to process fixed format files as they appear in a specified directory. It's also possible to use it with Excel files with some additional preparation. These instructions describe the additional steps required to use an Excel worksheet as a source.

For a complete example of using the File Listener, see *Loading a Target Table Using the File Listener* on page 594. For further explanation of the attributes described, see *File Listener Attributes* on page 514.

Procedure: How to Create a Synonym for Use With the File Listener and Excel Files

1. Right-click an application directory, point to *New*, and click *Synonym via Synonym Editor*.

   A new synonym opens in the workspace.

   If the Properties panel is not visible, right-click the synonym name (for example baseapp/synonym01) and click *Properties*.

2. From the SUFFIX drop-down menu, select *Fixed Format Files*.
3. From the IOTYPE drop-down menu, select *BINARY*.
4. Right-click the synonym name, point to *Insert*, and click *Segment Manually*.

   A new segment named SEG1 and a new field named FIELD1 are added. SEG1 is selected.

5. In the Properties panel, from the Type drop-down menu, select *Multiple*.
6. From the DATA_ORIGIN drop-down menu, select *LISTENER*. Enter the following values:
   1. From the CONNECTION drop-down menu, select <*local*> or an [S]FTP server.
   2. In the DIRECTORY and NAME boxes, enter the location and name of the file(s) you want to listen for.
   3. In the EXTENSION box, enter *xlsx*.
   4. Enter values for PICKUP, DISCARD, POLLING, and TIMEOUT as required.
7. In the workspace, click *FIELD1*. Enter the following values in the Properties pane:
   1. In the FIELDNAME box, enter *FILENAME_FULL*.
   2. In the Usage Format box, enter *A512*.

      Expand the *DBMS/Source Data Attributes* section.

   3. In the ALIAS box, enter *INSTANCE_FULL*.
   4. In the ACTUAL box, enter *A512*.

Procedure: How to Add a Reference to an Existing Synonym for an Excel Worksheet

If you have a synonym for the Excel worksheet already in place, follow this procedure.

1. Right-click the *SEG1* segment, point to *Insert*, and click *Reference to Existing Synonym as Child*.

   The Insert Reference to Existing Synonym as Child dialog box opens.

2. Click the synonym that describes the Excel worksheets that you want to use as a source and click *Select*.

   The reference to the synonym is added and the segment name is selected.

3. In the Properties panel, from the POSITION drop-down menu, select *INSTANCE_FULL*.

4. From the Quick Access toolbar click the save icon. Enter a name for the synonym and click *Save*.

You can now use the synonym you created as a source in a Data Flow or a Direct Load Flow.

Reference: Prototype File Monitoring Synonym

A prototype synonym is provided as filemntr.mas for a user with the File Listener pickup strategy Monitor. To create this synonym, first follow the instructions outlined in *How to Create Sample Procedures and Data for File Listener* on page 62. Then, copy it to an application directory in your application path and edit the File Listener parameters as needed. At a minimum, edit the value for DIRECTORY to point to the application directory (or mapped application name) that you want to monitor. Edit the NAME and EXTENSION values to show the file name (or * for all names) and extensions that you want to monitor.

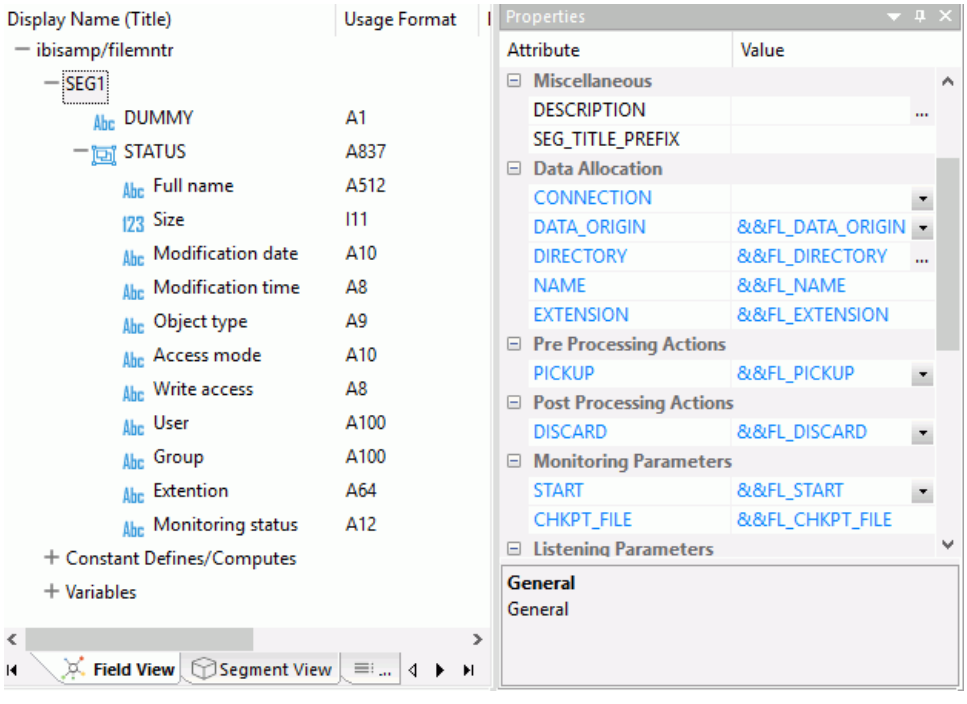
Note: These virtual fields can also be used with pickup strategy other than Monitor, except as noted.

This synonym contains fields with special values for the ALIAS, as described below.

- INSTANCE_FULL - The full path and name of the file. Cannot be used in conjunction with INSTANCE. Also available with IMMEDIATE.
- INSTANCE_SIZE - The size of the file in bytes. Also available with IMMEDIATE.
- INSTANCE_DATE - The date that the file was last modified. Also available with IMMEDIATE.
- INSTANCE_TIME - The time that the file was last modified. Also available with IMMEDIATE.
- INSTANCE_TYPE - The value File to indicate a file, or when DATA_ORIGIN is FILE, the value <DIR> to indicate a directory. This directory is only available with MONITOR. Otherwise, the value is always File.

- INSTANCE_ACCS - The access mode for the file. Available on UNIX systems only.
- INSTANCE_RDWR - Indicates if the file is READONLY. Available on UNIX systems only.
- INSTANCE_USER - The user that owns the file. Available on UNIX systems only.
- INSTANCE_GROUP - Group identification. Available on UNIX systems only.
- INSTANCE_MONS - Monitoring status. Blank for the initial run, then ADDED, DELETED, MODIFIED, or RENAMED to reflect changes. Only available with MONITOR.

The properties of the prototype synonym are shown in the following image.



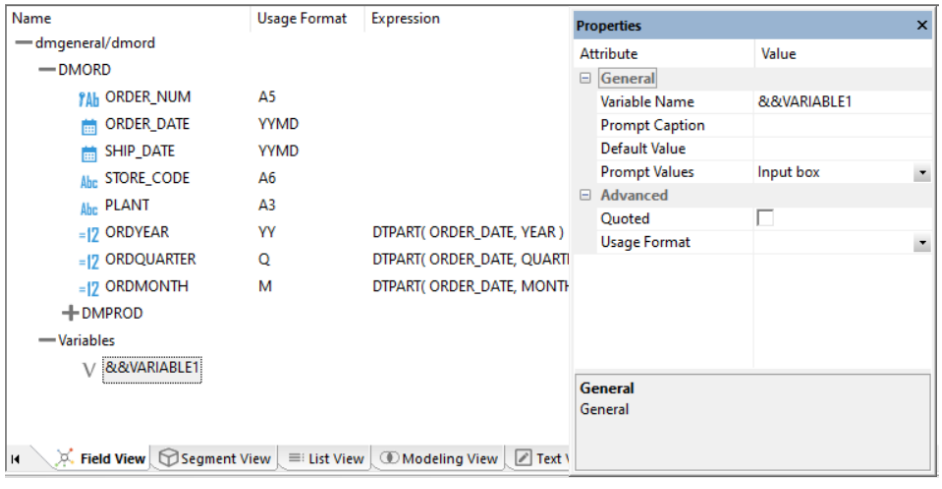
Using Variables in a Synonym

The Synonym Editor allows you to define variables in a synonym in order to specify synonym attributes, such as the connection name or the table name. For an example of how variables in a synonym can be used, see *Iterating Through Multiple Tables* on page 612.

Procedure: How to Add a Variable to a Synonym

1. Right-click an object in the Synonym Editor, point to *Insert*, and then click *Variable*.

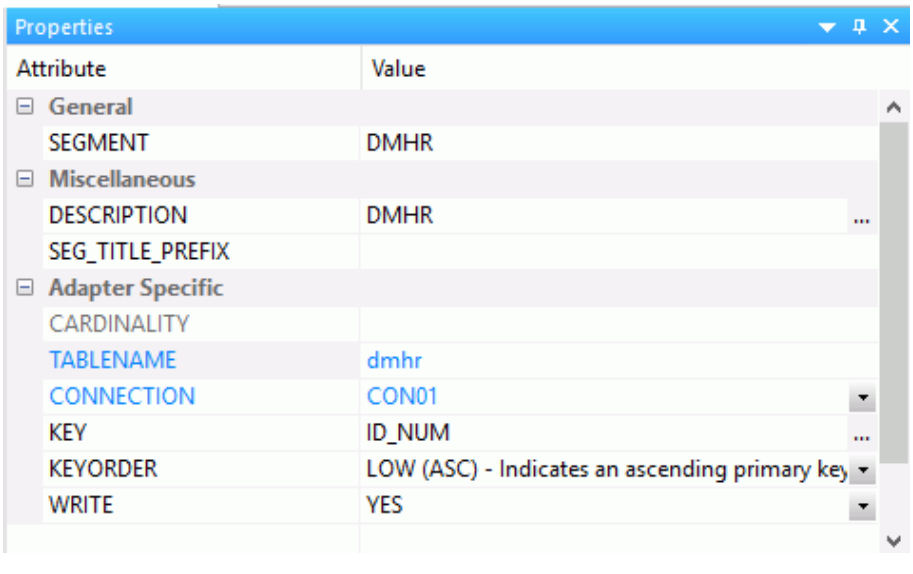
2. A new folder named Variables is added to the object tree, with &&VARIABLE1 as an entry. The Properties pane shows the attributes of the variable, as shown in the following image.



3. If desired, change the name of the variable and enter a default value in the appropriate fields.

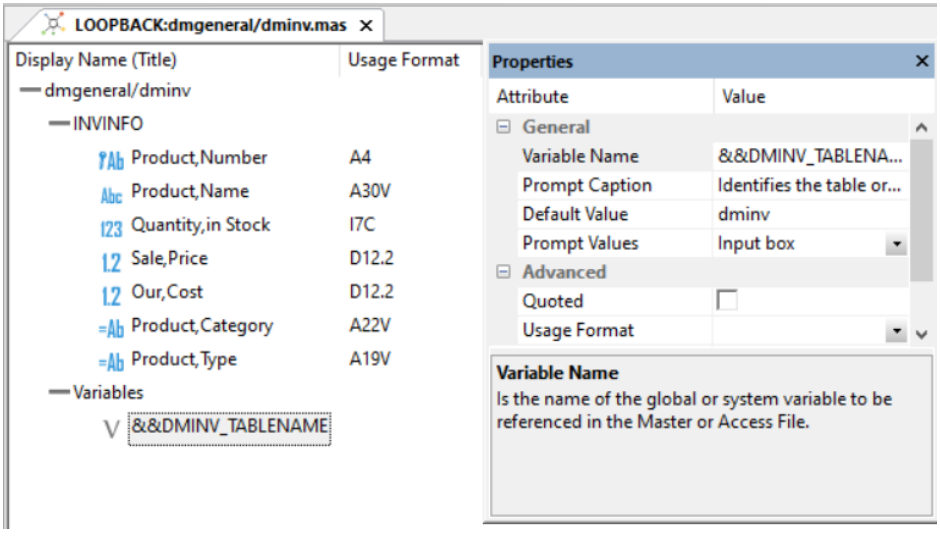
Procedure: How to Add a Variable for an Attribute

Attributes in a synonym that can be specified with a variable are identified with an attribute name in blue, as shown in the following image.



To add variables for an attribute:

1. Right-click the attribute in the Properties pane and click *Parameterize Value*.
2. Expand the Variables folder. A variable with a name that combines the synonym name with the variable name has been added.
3. Click the variable name to assign values, as shown in the following image.



Variables Attribute Summary

Variables in a synonym can have the following attributes:

Variable Name

The name of the variable. A double ampersand (&&) is automatically inserted in the name to indicate a global variable.

Prompt Caption

A description of the variable that replaces the variable name in the prompt.

Default Value

The default value for the variable. If no value is set, the default value is used.

Prompt Values

Provides a drop-down menu of optional prompt values for the variable. The options are Input box, Static list of values, Static list of values/captions, Synonym based dynamic list of values, and Procedure based dynamic list of values.

Quoted

When selected, single quotation marks are added around the assigned string for the variable. A single quotation mark within the string is converted into two single quotation marks.

Usage Format

The data type. The following options are available:

- Integer
- Character (fixed)
- Decimal

Use a Variable in a Synonym

Variables in a synonym are commonly used to represent a table name or connection name.

1. In a synonym for a relational database where variables have been established, click a segment to open the properties page that shows the Adapter Specific attributes.
2. In the fields for TABLENAME and CONNECTION, enter the names of the variables to be used.
3. Click *Save* to save your changes.

Creating and Enhancing Synonyms Using Modeling View

The Synonym Editor Modeling View tab allows you to create and enhance synonyms graphically. You can also use this view to create a single view of multiple data sources.

Procedure: How to Create Synonyms Using Modeling View

1. Right-click an application directory, point to *New*, and then click *Synonym via Synonym Editor*.
2. Right-click the new synonym, point to *Insert*, and click *Reference to Existing Synonym as Root* .
3. Select the synonym to use as the root, and click *Select*.
   Note: If the synonym selected has more than one segment, a dialog box opens. Click the entry segment you wish to use and click *OK*.
4. Click the *Modeling View* tab.
5. Right-click the new synonym in the workspace and point to *Insert*.
6. Add segments through one of the four methods listed.

7. Optionally, if you have multiple joined segments you can double-click the arrowed lines between tables to open the Join Editor dialog box and determine how the tables are linked.

Procedure: How to Enhance Synonyms Using the Modeling View

Use the Modeling View to enhance a synonym by adding a segment.

1. Open the synonym in the Synonym Editor by double-clicking it from the navigation pane.
2. Click the *Modeling View* tab.
3. Right-click a segment in the workspace and point to *Insert*.
4. Add segments through one of the methods listed:

   - **Reference to Existing Synonym as Child.** Adds a segment to the existing synonym as a reference to another synonym. Only the name of the synonym is added. If the referenced synonym is changed, the new version is used.
   - **Reference to Existing Synonym as Root.** Inserts reference to selected table and any tables it references.
   - **Segment via Metadata Import.** Enables you to add segments by using the Create Synonym tool. This tool creates a synonym and includes it as a segment in the synonym from which the tool was launched.
   - **Segment Manually.** Enables you to add segments manually to the current synonym.

     Note: You would only use this approach if you are coding a new synonym by hand, as you would for a FOCUS data source.

5. Optionally, you can double-click the arrowed lines between tables to open the Join Editor dialog box and determine how the tables are linked.

   Tip: You may also right-click a file icon and click *Join Properties* from the context menu.

   In addition, when creating a cluster join, there are two additional options for linking tables. Join All (default) is shown by a double arrow, and Join Unique is shown as a single arrow.

   Join options are not available for FOCUS files.

Reference: Modeling View - Segment Shortcut Menu

When you right-click the title bar of a segment in the Synonym Editor Modeling View tab, a shortcut menu opens.

The shortcut menu has the following options:

Expand/Collapse

Changes the view from a file icon to a table view, which enables you to see columns, sample data, and sample data parent keys.

Tip: You may also double-click a file icon to open the table view. Double-click the table to close the table view, or click the X button from the toggle toolbar to close.

Properties

Opens the Properties pane for the selected segment.

Insert

Has the following options:

- **Reference to Existing Synonym as Child.** Adds a segment to the existing synonym as a reference to another synonym. Only the name of the synonym is added. If the referenced synonym is changed, the new version is used.
- **Reference to Existing Synonym as Root.** Inserts reference to selected table and any tables it references.
- **Segment via Metadata Import.** Enables you to create a new synonym through the Create Synonym tool, and add it to the current synonym.
- **Segment Manually.** Adds a segment to the current synonym that must be coded manually.

Data Profiling

Provides the data characteristics for the columns of a synonym.

Sample Data

Shows a sample of the data associated with the synonym.

Delete

Deletes the segment.

Edit All Properties

Opens the Properties pane for the segment.

Assign SCD Columns

Allows you to assign columns to a SCD type by using drag-and-drop.

Adding Indexes to a Synonym

You can create indexes in a synonym by adding them in the Synonym Editor. When you create a table by right-clicking a synonym, pointing to *Data Management*, and clicking *Recreate DBMS table*, the indexes are created in the relational data base.

When you point to *New* and click *Synonym* to create a synonym for a relational database table with more indexes than just the primary key, they will be included in the synonym.

Additional indexes are useful when you anticipate using a column in a JOIN operation, a WHERE condition, or as an argument to DB_LOOKUP. You can add multiple indexes. Each index can be on one or more columns. You can specify if the index is sorted, and whether it is a unique index.

Example: Adding an Index to a Synonym

This example uses the sample synonym ibisamp/dmreps that has a primary key on column REPNO and a unique index on that column. It shows how to add an index on the column EMPID, and another index on the FNAME and LNAME columns that does not have to be unique.

1. From the ibisamp application directory, double-click the *dmreps* synonym to open it in the Synonym Editor.
2. Right-click the segment name *DMREPS*, point to *Insert*, and then click *Index*.

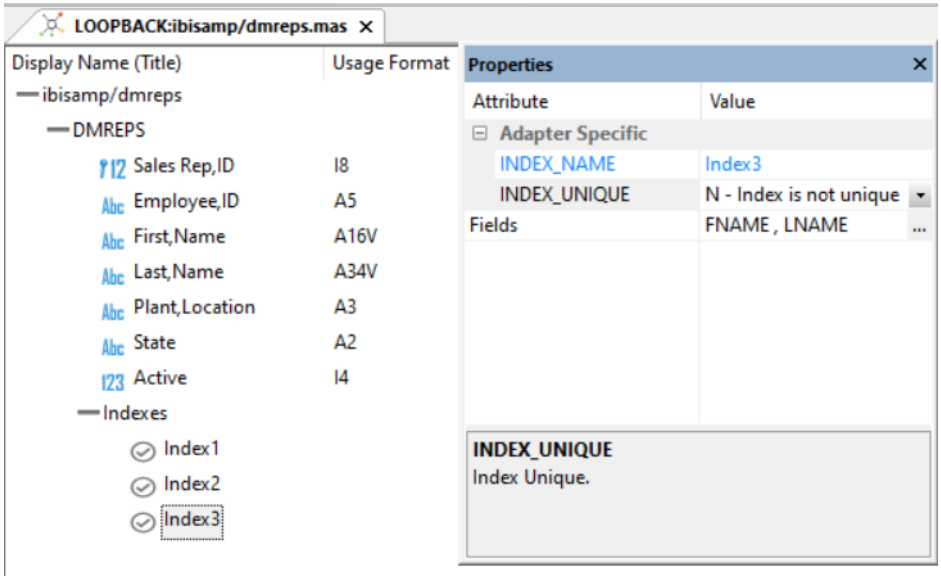
   The Index Fields Editor dialog box opens. The field Sales Rep,ID (REPNO) is already selected because it is the key column of the table.
3. Click *OK*.
4. A new folder named Indexes appears in the tree with Index1 defined for REPNO. Right-click the *Indexes* folder, point to *Insert*, and then click *Index*.

   The Index Fields Editor dialog box opens.
5. Double-click *Employee,ID (EMPID)* in the Available Fields pane to add it to the Selected Field(s) pane.
6. Click *OK*.
7. Right-click the *Indexes* folder, point to *Insert*, and then click *Index*.

   The Index Fields Editor dialog box opens.
8. Double-click *First,Name (FNAME)* and *Last,Name (LNAME)* to add them to the Selected Field(s) pane.
9. Click *OK*.

10. In the Properties pane for Index3, select *N – Index is not unique* in the INDEX_UNIQUE drop-down menu.

    Your synonym should look like the following image.



Adding Virtual Columns (DEFINE) in a Synonym

You can create transformations in a synonym by adding a virtual column, also known as a DEFINE. This creates a transformation that appears with the data source list of columns.

A virtual column in the synonym is useful in the following situations:

- **When you want to create static transformations that are constant through an organization.** For example, you might want to create a column of phone numbers with the dashes removed or a list of Department codes decoded (such as, 1 for Accounting, 2 for Marketing, and so on).
- **When you want all data flows to use a set of common business rules.** For example, you might want to set a definition like PROFIT = REVENUE - EXPENSES.
- **When you want to simplify data flow maintenance.** If the PROFIT expression noted above ever changes, it only needs to be edited at the source Server.

A virtual column in the synonym can also be used in the following situations, as an alternative to using a source transformation:

- **When you want to filter information based on the results of a transformation.** For example, you might want to filter based on the size of PROFIT.
- **When you want to group by the results of a calculation.** For example, the table has an order date column, and you want to group on the year extracted from that column.
- **When you want to create transformations based on user routines that are located on the source machine.** Specialized in-house routines, tested over time, may be required for your warehousing application, or the Reporting Server and your data mart may be located on a UNIX platform. Business rules may dictate that a z/OS COBOL program be executed for each retrieved row. You can exploit this COBOL program, using a virtual column.

It is often useful to prototype the transformations with a limited number of rows, and then place them in the source definition after analyzing your performance.

Note: If you have a data flow that uses the synonym in which you are creating a virtual column, you should right-click the synonym's application directory and select *Refresh* to use the new column.

A virtual column can contain an expression, a constant, or a column name.

- If the virtual column is a complex expression, you can create the expression with the Transformation Calculator or just type it into the Expression field.
- If the virtual column is a simple expression, such as a constant value, you can type the value in the Expression field.

After creating a virtual column, you can test it by right-clicking the synonym and clicking *Sample Data*. Sample data appears in a separate dialog box.

Procedure: How to Create a Virtual Column in a Synonym

To create a virtual column in a synonym:

1. From an application directory in the navigation pane, double-click a synonym, or right-click it and click *Open*.

   The Synonym Editor opens.

2. Right-click a segment (or column), point to *Insert*, and then click *Define*.

   The Define Calculator opens.

3. From the Define Calculator, type a name for the column in the Name input field, or use the default define name.

4. You may enter a descriptive title for the virtual column (DEFINE) in the Title input field.

   Tip: From the Synonym Editor, click the browse *(...)* button at the right of the TITLE and DESCRIPTION value fields to specify multiple language titles.

5. Use the Expression tab and the calculator buttons to build the expression for the virtual column (DEFINE).

   or

   Use the Relational Expression tab to build the expression.

   a. Click the *Add New Row* icon. From the Relational Expression tab, use the drop-down lists to select the filter Field (column), Relation, and Type.

      Note: Parameters are not supported.

   b. Click the browse *(...)* button at the right of the Value input field.

      The Value Selection dialog box opens.

   c. Select from the available values and use the arrows to add or remove values.

   d. Click *OK* to close the Value Selection dialog box and return to the Relational Expression tab.

      The expression is added to the value field.

   e. To add another filter, double-click a column or variable from the Columns/Variables tab on the right side of the Virtual Column Calculator.
      The filter is added to the Relational Expression tab, where you can add the expression value.

   f. To delete an expression, click in the row number column of the expression that you are deleting, right-click, and select *Delete selected row(s)*. The expression is removed from the Relational Expression tab.

6. You may select the *Check Expression* and *Sample Data* buttons, located on the top right of the Virtual Column Calculator, to verify that the expression is valid and to view sample data for the filter.

7. Click *OK* to close the Define Calculator and return to the Synonym Editor.

   Note: To edit the Define, Title, or Expression, you may do so directly from the Synonym Editor, or you may click the browse *(...)* button at the right of the EXPRESSION value field to relaunch the Define Calculator.

8. If no columns from the synonym are used in the expression or have been defined, you can use the WITH option to identify the logical home of the defined calculation. You can also use the WITH option to move the logical home for the virtual column to a lower segment than it would otherwise be assigned (for example, to count instances in a lower segment).

   Tip: You can click and drag the DEFINE field and move it to a different segment in the Modeling View tab, which also changes the segment association.

9. Specify the Missing Data options for columns that allow null data. You can allow all missing data.
10. Click *Save* from the Quick Access Toolbar to save the synonym.
11. To close the Synonym Editor, click the control button in the upper-right corner.

Viewing Data Profiling Characteristics

Data Profiling provides data characteristics for the columns in a synonym. You can display the characteristics for all the columns in a synonym or segment, or for an individual column.

For alphanumeric columns, Data Profiling provides the segment, format, count of distinct values, total count, patterns count, maximum, minimum, and average length, minimum and maximum values, and number of nulls. Patterns count shows the number of patterns found in each alphanumeric column.

For numeric columns, Data Profiling provides the segment, format, count of distinct values, total count, maximum, minimum, and average values, and number of nulls.

Data Profiling for an individual column provides access to Statistics, Patterns, Values, and Outliers reports.

Data Profiling a Synonym or Segment

Data Profiling provides information on all the columns in a synonym or segment. You can also drill down to the Values or Patterns reports for an individual column from a synonym or segment Data Profiling report.

Note: Data Profiling is also available from the navigation pane by right-clicking a synonym, pointing to *Data Profiling*, and then clicking *Statistics*.

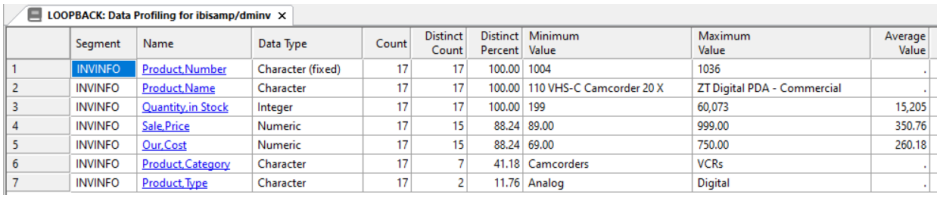
Procedure: How to View Data Profiling for a Synonym or Segment

To view the Data Profiling information for a synonym or segment:

1. Open the synonym in the Synonym Editor by double-clicking it from the navigation pane or right-clicking it and clicking *Open*.

   The Synonym Editor opens to the Field View tab.

2. Right-click the synonym or segment name, point to *Data Profiling*, and then click *Statistics*.


LOOPBACK: Data Profiling for ibisamp/dminv

	Segment	Name	Data Type	Count	Distinct Count	Distinct Percent	Minimum Value	Maximum Value	Average Value
1	INVINFO	Product.Number	Character (fixed)	17	17	100.00	1004	1036	.
2	INVINFO	Product.Name	Character	17	17	100.00	110 VHS-C Camcorder 20 X	ZT Digital PDA - Commercial	.
3	INVINFO	Quantity.in Stock	Integer	17	17	100.00	199	60,073	15,205
4	INVINFO	Sale.Price	Numeric	17	15	88.24	89.00	999.00	350.76
5	INVINFO	Our.Cost	Numeric	17	15	88.24	69.00	750.00	260.18
6	INVINFO	Product.Category	Character	17	7	41.18	Camcorders	VCRs	.
7	INVINFO	Product.Type	Character	17	2	11.76	Analog	Digital	.

The Data Profiling information displays in the workspace.

You may use the Report tab in the ribbon to view server messages, print the report, copy data as text, and export the report.

3. Optionally, you can click a column name or patterns count (for alphanumeric columns) to drill down to the Values or Patterns reports, respectively.

   For pattern analysis, a 9 represents a digit, an A represents any uppercase letter, and an a represents any lowercase letter. All printable special characters are represented by themselves, and unprintable characters are represented by an X.

Note: Data Profiling is also available from the navigation pane by right-clicking a synonym.

Procedure: How to View Key Analysis for a Synonym or Segment

Key Analysis provides a report that shows which columns in a data source can be used individually, or in combination, to uniquely identify a row. The columns identified in this report are candidates for key columns.

Note: Key Analysis is also available from the navigation pane by right-clicking a synonym, pointing to *Data Profiling*, and then clicking *Key Analysis*.

To view key analysis for a synonym or segment:

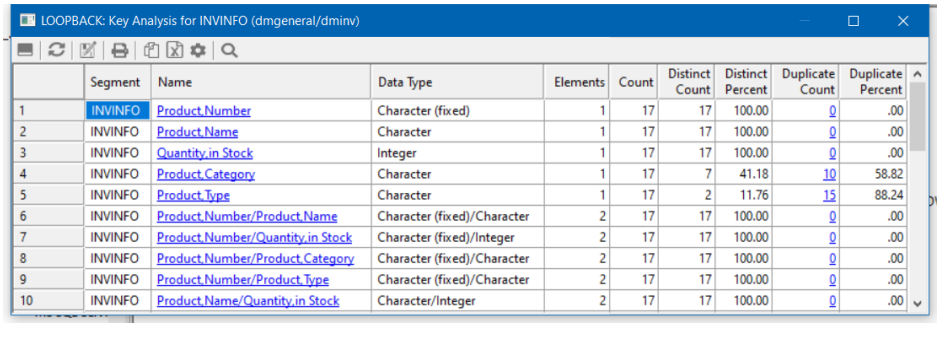
1. Open the synonym in the Synonym Editor.
2. Right-click the synonym or segment name, point to *Data Profiling*, and then click *Key Analysis*.

   The Key Analysis dialog box opens.

3. Select the columns that you would like to analyze as potential keys to the source table and click *Analyze*.

   The Key Analysis report for that synonym box opens.

4. If you selected a segment name, the report opens as shown in the following image. Note that all columns in the segment will be selected.



	Segment	Name	Data Type	Elements	Count	Distinct Count	Distinct Percent	Duplicate Count	Duplicate Percent
1	INVINFO	Product.Number	Character (fixed)	1	17	17	100.00	0	.00
2	INVINFO	Product.Name	Character	1	17	17	100.00	0	.00
3	INVINFO	Quantity.in Stock	Integer	1	17	17	100.00	0	.00
4	INVINFO	Product.Category	Character	1	17	7	41.18	10	58.82
5	INVINFO	Product.Type	Character	1	17	2	11.76	15	88.24
6	INVINFO	Product.Number/Product.Name	Character (fixed)/Character	2	17	17	100.00	0	.00
7	INVINFO	Product.Number/Quantity.in Stock	Character (fixed)/Integer	2	17	17	100.00	0	.00
8	INVINFO	Product.Number/Product.Category	Character (fixed)/Character	2	17	17	100.00	0	.00
9	INVINFO	Product.Number/Product.Type	Character (fixed)/Character	2	17	17	100.00	0	.00
10	INVINFO	Product.Name/Quantity.in Stock	Character/Integer	2	17	17	100.00	0	.00

The report shows:

Segment

The selected segment.

Name

Name of the segment.

Format

The format of each column.

Elements

The number of elements (columns) shown.

Count

The number of rows.

Distinct Count

The number of distinct rows.

Distinct Percent

The percentage of rows that are distinct. This value must be 100% for a combination of columns to be used as key.

Duplicate Count

The number of duplicate values.

Duplicate Percent

The percentage of duplicate values. This value must be 0% for a combination of columns to be used as key.

Example: Select Table DMHR and All of the Columns EXCEPT ID_NUM and Capture the Output

By default, the report is sorted by the number of elements, so the first rows in the report show one element each. This enables you to determine if any single column could be used by itself as a key. The report then shows all combinations of two columns, three columns, and so on.

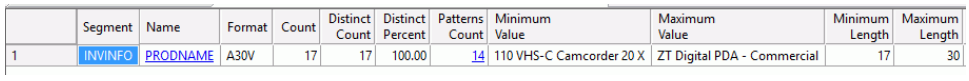
To see the values in the report, right-click on any row.

The duplicate rows option shows all duplicate values, which prevent the desired column combination from being used as a key.

Data Profiling a Single Column

Data Profiling for an individual column provides access to nine reports:

- **Statistics.** Lists counts, percents, limits, distinct values, and information about null values in a field or a segment, as shown in the following image.

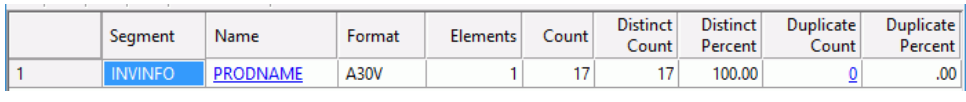


	Segment	Name	Format	Count	Distinct Count	Distinct Percent	Patterns Count	Minimum Value	Maximum Value	Minimum Length	Maximum Length
1	INVINFO	PRODNAME	A30V	17	17	100.00	14	110 VHS-C Camcorder 20 X	ZT Digital PDA - Commercial	17	30

  For alphanumeric columns, the Statistics report provides the segment, format, count of distinct values, total count, patterns count, maximum, minimum, and average length, minimum and maximum values, and number of nulls.

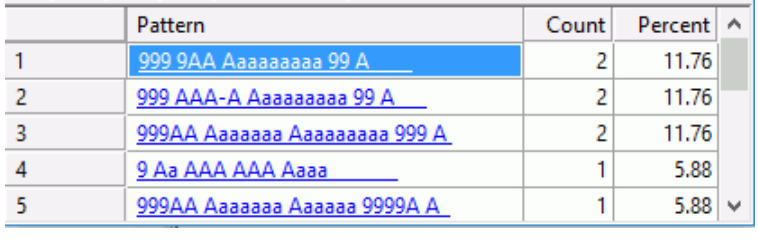
  For numeric columns, the Statistics report provides the segment, format, count of distinct values, total count, maximum, minimum, and average values, and number of nulls.

- **Key Analysis.** Lists counts and distinct and duplicate counts and percents for a single column, as shown in the following image.



	Segment	Name	Format	Elements	Count	Distinct Count	Distinct Percent	Duplicate Count	Duplicate Percent
1	INVINFO	PRODNAME	A30V	1	17	17	100.00	0	.00

- **Patterns** Shows patterns of letters, digits, and special characters, as well as counts and their percents, as shown in the following image. This report is only available for alphanumeric columns.

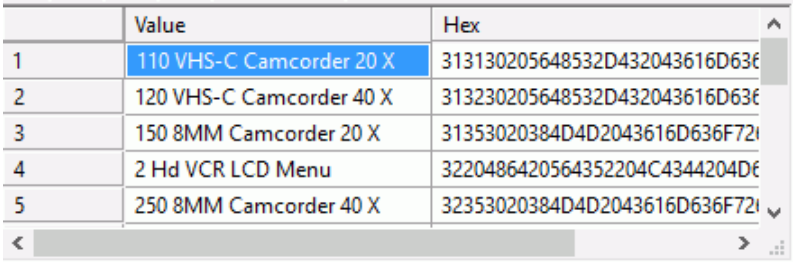


	Pattern	Count	Percent
1	999 9AA Aaaaaaaaa 99 A	2	11.76
2	999 AAA-A Aaaaaaaaa 99 A	2	11.76
3	999AA Aaaaaaa Aaaaaaaaa 999 A	2	11.76
4	9 Aa AAA AAA Aaaa	1	5.88
5	999AA Aaaaaaa Aaaaaa 9999A A	1	5.88

  Optionally, you can click the pattern to drill down to the values for the rows containing those patterns.

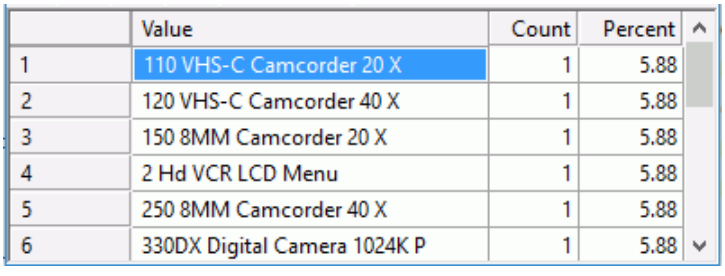
  For pattern analysis, a 9 represents a digit, an A represents any uppercase letter, and an a represents any lowercase letter. All printable special characters are represented by themselves, and unprintable characters are represented by an X.

- **Hex View.** Shows the hexadecimal value of any field, as shown in the following image.



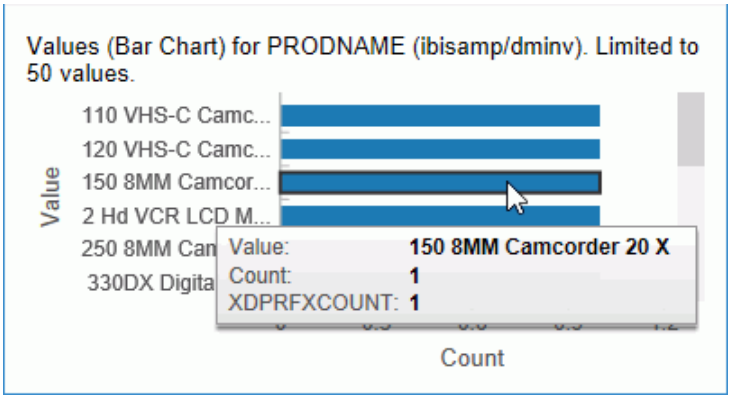
	Value	Hex
1	110 VHS-C Camcorder 20 X	313130205648532D432043616D636
2	120 VHS-C Camcorder 40 X	313230205648532D432043616D636
3	150 8MM Camcorder 20 X	31353020384D4D2043616D636F72
4	2 Hd VCR LCD Menu	3220486420564352204C4344204D6
5	250 8MM Camcorder 40 X	32353020384D4D2043616D636F72

- **Frequent Values.** Shows values and their percents, as shown in the following image.

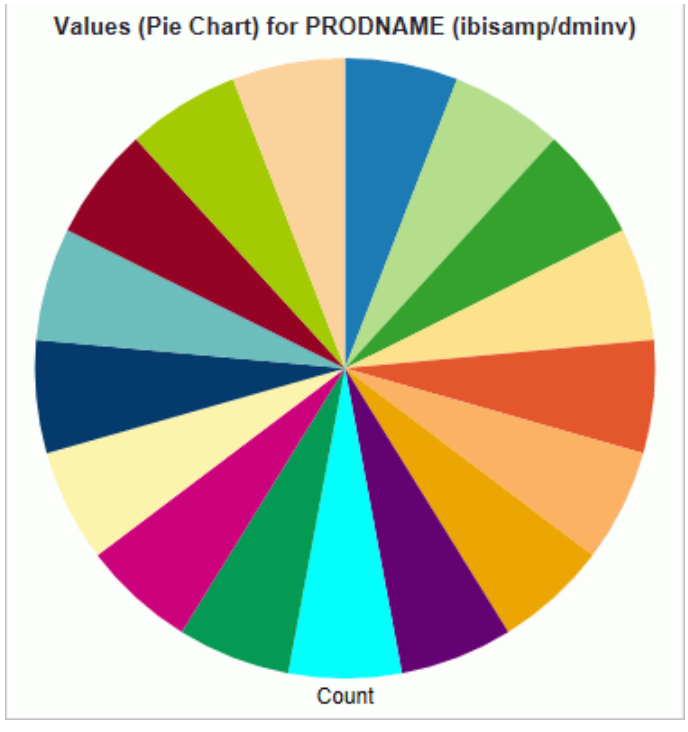


	Value	Count	Percent
1	110 VHS-C Camcorder 20 X	1	5.88
2	120 VHS-C Camcorder 40 X	1	5.88
3	150 8MM Camcorder 20 X	1	5.88
4	2 Hd VCR LCD Menu	1	5.88
5	250 8MM Camcorder 40 X	1	5.88
6	330DX Digital Camera 1024K P	1	5.88

- **Frequent values chart.** shows a horizontal bar graph of the field values, as shown in the following image. The tooltip for each bar also shows the count of instances of the field.



- **Frequent values pie chart.** Shows a pie chart of the field values, as shown in the following image.



The tooltip for each bar also shows the count of instances of the field and the percent of the total.

- **Duplicate Values.** Lists the values that have multiple instances, the count of instances, and the corresponding percent, as shown in the following image.

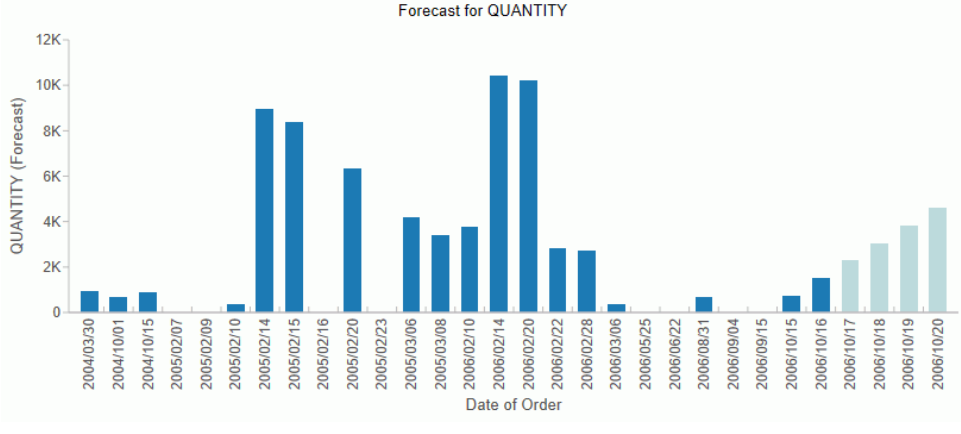
	Value	Count	Percent
1	2005/02/14 00:00	17	16.35
2	2005/02/15 00:00	15	14.42
3	2005/02/20 00:00	11	10.58
4	2005/03/06 00:00	10	9.62
5	2005/03/08 00:00	2	1.92
6	2006/02/10 00:00	3	2.88
7	2006/02/14 00:00	13	12.50

- **Outliers.** Shows high and low values with a count of instances that fall outside the bulk of the data for a field, as shown in the following image.

	High	Count	Low	Cour
1	ZT Digital PDA - Commercial	1	110 VHS-C Camcorder 20 X	
2	ZC Digital PDA - Standard	1	120 VHS-C Camcorder 40 X	
3	R5 Micro Digital Tape Recorder	1	150 8MM Camcorder 20 X	
4	QX Portable CD Player	1	2 Hd VCR LCD Menu	
5	DVD Upgrade Unit for Cent. VCR	1	250 8MM Camcorder 40 X	
6	Combo Player - 4 Hd VCR + DVD	1	330DX Digital Camera 1024K P	
7	AR3 35MM Camera 10 X	1	340SX Digital Camera 65K P	

Note: Outliers produce a maximum of 10 highest and lowest distinct values, if they exist.

- **Forecast Chart.** Shows the values used for both the forecast and forecast values, as shown in the following image. This report is only available for numeric columns.



These reports are available by right-clicking a column in the Synonym Editor and pointing to *Data Profiling*.

Procedure: How to View a Data Profiling Report

To view a Data Profiling report for a single column:

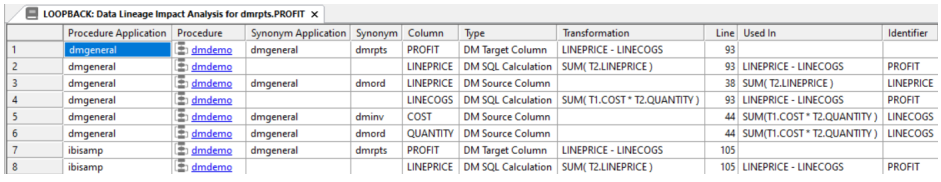
1. Open the synonym in the Synonym Editor by double-clicking it from the navigation pane, or right-clicking it and clicking *Open*.

   The Synonym Editor opens to the Field View tab.

2. Right-click a column, point to *Data Profiling*, and then click one of the available reports.

Reference: **Data Profiling Statistics Report Columns**

The Statistical Data Profiling information displays in the workspace, as shown in the following image.



LOOPBACK: Data Lineage Impact Analysis for dmrpts.PROFIT

	Procedure Application	Procedure	Synonym Application	Synonym	Column	Type	Transformation	Line	Used In	Identifier
1	dmgeneral	dmdemo	dmgeneral	dmrpts	PROFIT	DM Target Column	LINEPRICE - LINECOGS	93		
2	dmgeneral	dmdemo			LINEPRICE	DM SQL Calculation	SUM(T2.LINEPRICE)	93	LINEPRICE - LINECOGS	PROFIT
3	dmgeneral	dmdemo	dmgeneral	dmord	LINEPRICE	DM Source Column		38	SUM(T2.LINEPRICE)	LINEPRICE
4	dmgeneral	dmdemo			LINECOGS	DM SQL Calculation	SUM(T1.COST * T2.QUANTITY)	93	LINEPRICE - LINECOGS	PROFIT
5	dmgeneral	dmdemo	dmgeneral	dminv	COST	DM Source Column		44	SUM(T1.COST * T2.QUANTITY)	LINECOGS
6	dmgeneral	dmdemo	dmgeneral	dmord	QUANTITY	DM Source Column		44	SUM(T1.COST * T2.QUANTITY)	LINECOGS
7	ibisamp	dmdemo	dmgeneral	dmrpts	PROFIT	DM Target Column	LINEPRICE - LINECOGS	105		
8	ibisamp	dmdemo			LINEPRICE	DM SQL Calculation	SUM(T2.LINEPRICE)	105	LINEPRICE - LINECOGS	PROFIT

The report shows:

Segment

The name of the segment in the synonym. For a cluster synonym, each segment can refer to a separate table; a segment can also refer to a group of columns.

Name

The column name. If you click the name, another report opens for that column showing each unique value and a count of the number of times the value appears and the percentage

Format

The FOCUS format of the column.

Count

The number of values for the field.

Distinct Count

The number of unique values for the column.

Distinct Percent

The percentage of values that are unique.

Patterns Count

The number of unique patterns (combinations of letters and digits) that are found.

Average Value

Only for numeric fields.

Minimum, Maximum, and Average Length

Only for character fields.

Nulls Count
The number of values that are missing.

Nulls Percent
The percentage of values that are missing.

Duplicate Count

The number of values that are not unique.

Duplicate Percent

The percentage of values that are not unique.

The list of columns displayed can be selected from the ribbon by clicking *Customize Columns*. Clear any that are not desired. The following optional columns can also be displayed:

Median
The value where half the values are less and half the values are more.

Mode
The most commonly occurring value.

Optionally, you can click a column name or patterns count (for alphanumeric columns) to drill down to the Values or Patterns reports, respectively.

Note: Data profiling requires reading every row and column of the source table. Use caution with large data sources.

Viewing Data Lineage for a Column

Data lineage provides the ability to see how a column was derived. It traces the column back to the data source(s) it came from, breaking down all of its components along the way.

Data lineage is available from the Synonym Editor. It can show, for example, that a target column is the result of an SQL calculation that uses source columns from other synonyms.

Procedure: How to View the Data Lineage of a Column

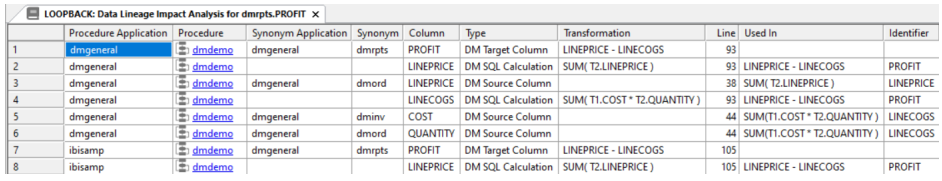
To view the data lineage information for a column:

1. Open the synonym in the Synonym Editor by double-clicking it in the navigation pane, or right-clicking it and clicking *Open*.

   The Synonym Editor opens to the Field View tab.

2. Right-click a column and click *Data Lineage*.

The lineage information displays in the workspace, as shown in the following image.



LOOPBACK: Data Lineage Impact Analysis for dmrpts.PROFIT

	Procedure Application	Procedure	Synonym Application	Synonym	Column	Type	Transformation	Line	Used In	Identifier
1	dmgeneral	dmdemo	dmgeneral	dmrpts	PROFIT	DM Target Column	LINEPRICE - LINECOGS	93		
2	dmgeneral	dmdemo			LINEPRICE	DM SQL Calculation	SUM(T2.LINEPRICE)	93	LINEPRICE - LINECOGS	PROFIT
3	dmgeneral	dmdemo	dmgeneral	dmord	LINEPRICE	DM Source Column		38	SUM(T2.LINEPRICE)	LINEPRICE
4	dmgeneral	dmdemo			LINECOGS	DM SQL Calculation	SUM(T1.COST * T2.QUANTITY)	93	LINEPRICE - LINECOGS	PROFIT
5	dmgeneral	dmdemo	dmgeneral	dminv	COST	DM Source Column		44	SUM(T1.COST * T2.QUANTITY)	LINECOGS
6	dmgeneral	dmdemo	dmgeneral	dmord	QUANTITY	DM Source Column		44	SUM(T1.COST * T2.QUANTITY)	LINECOGS
7	ibisamp	dmdemo	dmgeneral	dmrpts	PROFIT	DM Target Column	LINEPRICE - LINECOGS	105		
8	ibisamp	dmdemo			LINEPRICE	DM SQL Calculation	SUM(T2.LINEPRICE)	105	LINEPRICE - LINECOGS	PROFIT

Chapter 11

Advanced Tips and Techniques

This chapter describes several advanced techniques for Data Migrator.

The examples used to illustrate these techniques use sample databases and data, which you must create. The synonyms and sample data are created in the ibisamp directory. This directory is automatically installed when you install the server, and is automatically added to the server path.

In this chapter:

- Overview of Examples
- Creating Sample Data and Metadata
- Horizontal Partitioning with Multi-Table Load
- Using the Decode Function
- Generating a Surrogate Key
- Generating a Surrogate Key Using Slowly Changing Dimensions
- Mapping to a Sequential File of a Specific Format
- Creating and Using Reusable Transformations
- Using Multiple Targets to Create a Single File With Multiple Records Types
- Creating a Target File With Multiple RECTYPEs
- Combining Multiple Source Rows in a Single Target Row
- Using Lookups to Create Multi-Table Flows Without Joins
- Record Logging
- Using Variables in Data Flows
- Creating a Data Flow Using a Target Based on a Predefined XML Schema
- Loading a Target Table Using the File Listener
- Sending a Target File to an FTP or SFTP Server
- Loading a Star Schema
- Iterating Through Multiple Tables
- Using Subqueries
- Loading Images Into a Database Table
- Updating Values in a Table

Overview of Examples

The following table lists and describes some of the advanced techniques available to you with Data Migrator. Step-by-step instructions are provided for each sample flow. The sample flows include:

Flow	Applies to Topic...
DMSPLIT	*Horizontal Partitioning with Multi-Table Load* on page 544
DMCODE	*Using the Decode Function* on page 548
DMSURRG	*Generating a Surrogate Key* on page 551
DMSURSCD	*Generating a Surrogate Key Using Slowly Changing Dimensions* on page 557
DMSEQ	*Mapping to a Sequential File of a Specific Format* on page 561
DMREUSE	*Creating and Using Reusable Transformations* on page 563
DMRCTYP	*Creating a Target File With Multiple RECTYPEs* on page 571
DMPIVOT	*Combining Multiple Source Rows in a Single Target Row* on page 575
DMLKUPS	*Using Lookups to Create Multi-Table Flows Without Joins* on page 579
DMDUPLG	*Record Logging* on page 583
DMVAR	*Using Variables in Data Flows* on page 586
DMORDXML	*Creating a Data Flow Using a Target Based on a Predefined XML Schema* on page 590
DMFLORD	*Loading a Target Table Using the File Listener* on page 594

The sample files listed and described in the following table are used in the examples.

Sample File	Description
dmcomp.mas	Sample - Stores

Sample File	Description
dmhr.mas	Sample - Human Resources
dmhrinp.mas	Sample target - Human Resources Input
dminv.mas	Sample - Inventory
dminva.mas	Sample target - Inventory Analog
dminvd.mas	Sample target - Inventory Digital
dmlocs.mas	Sample target - Plant locations
dmord.mas	Sample - Orders
dmreps.mas	Sample target - Sales Reps
dmrpts.mas	Sample target - Reporting
dmsale.mas	Sample - Sales Reps
dmstart.mas	Sample target - Employee Start records
dmterm.mas	Sample target - Employee Termination records
dmordfl.mas	Sample - Orders (fixed format file)
dmordf.mas	Sample target - Orders

Creating Sample Data and Metadata

Before you create and run the data flows discussed in this chapter, you must create the sample data that you will be using.

To create the sample data from the Data Management Console, see *How to Create Sample Procedures and Data* on page 59.

To create the sample data from the Web Console, see the *TIBCO WebFOCUS® Reporting Server Administration* manual or the Web Console online help.

Note: You should set your Run Options for *Maximum number of rows for test reports* to 100 in the Run Options section of the Options dialog box so that the sample flows will retrieve data using any database.

Tip: You can build your flows in any application directory in the servers search path, including the ibisamp directory. However, you may find it useful to create a dedicated sample application directory called *mysamp* (or another name) and add it to the server search path, following the directions in *Create Sample Data and Metadata and Create an Application Directory* on page 419.You can then build all of the flows in that directory, using sample data and metadata, as required, from the ibisamp directory.

Horizontal Partitioning with Multi-Table Load

You can use Data Migrator to build a flow that extracts data using a single pass of the data and loads multiple data targets. This can be useful if, for example, you need to create separate databases for each department or region in your enterprise and you can split the extracted data based on the value of a department code or region code. The term for splitting the data based on the value of a column in the extracted data is horizontal partitioning. Horizontal partitioning of the data is just one useful way to take advantage of the ability to load multiple data targets in Data Migrator.

Implement horizontal partitioning in Data Migrator by building a data flow using multiple data targets together with VALIDATE to extract data from a data source, and split the output to two different data targets based on the value of an input column. This can be accomplished in one pass of the data.

To do this, create a data flow with one or more data sources and multiple data targets. Each data target has its own VALIDATE statement based on the value of a particular column. When the data target is loaded, only records with the appropriate values are accepted; all other records are rejected.

In the following example, DMINV, an inventory table, is the data source, and PRODTYPE is the input column used to split the output into two data targets. Records containing a value of Analog for PRODTYPE are loaded into a data target named DMINVA. Records containing a value of Digital for PRODTYPE are loaded into a data target named DMINVD.

Refer to the sample data flow DMSPLIT for the complete example.

Note: DMSPLIT in the ibisamp directory writes to FOCUS/FDS data targets. If you installed the samples for a different adapter, you will need to open and resave DMSPLIT before running it.

Example: Creating a Data Flow for Horizontal Partitioning

Right-click the flow canvas, select *Flow Properties*, and deselect *Optimize Load*. Then close the Properties panel.

In the DMC:

1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the workspace with the SQL object displayed.
2. Drag the data source object *DMINV* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Right-click the SQL object and click *Column Selection*.

   The Column Selection window opens.
4. Select each column in the Available Columns list and click the double arrow to move them into the Selected Columns list. Click *OK*.

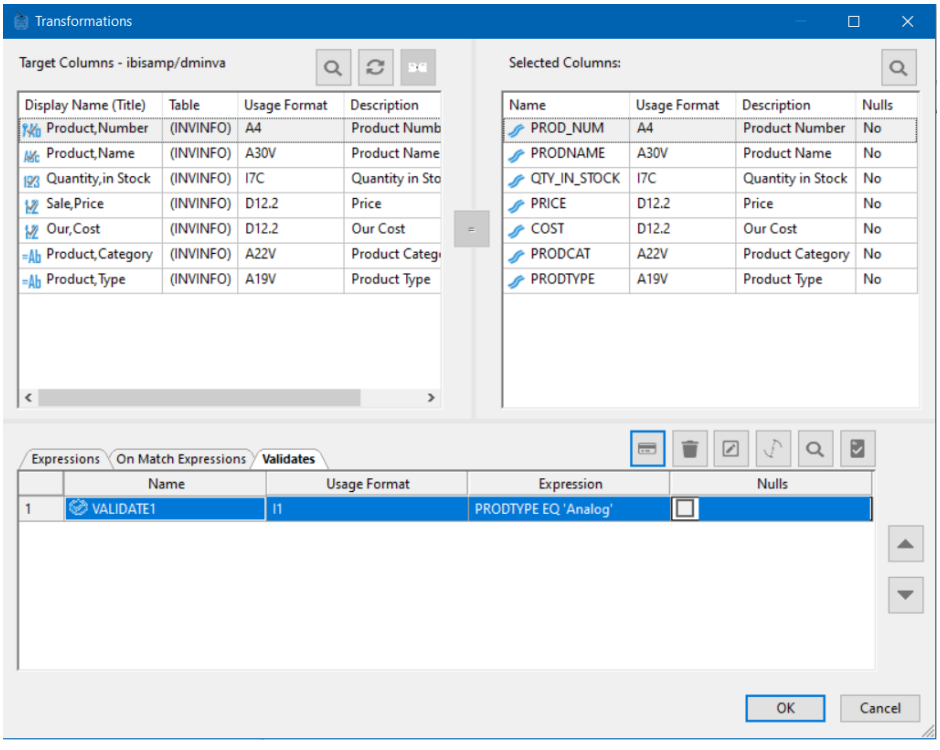
   Tip: To select all columns, click any cell and press Ctrl + A.
5. Drag the data target object *DMINVA* from the ibisamp application directory into the workspace, to the right of the SQL object.
6. Right-click the target object *DMINVA*, and click *Properties*.

   The Properties dialog box opens.
7. If the flow may be rerun, you should clear the target, using the *Prior to Load Option*. For a relational target, select *Delete all rows from table* or *Truncate Table*. For a FOCUS/FDS target, select *Drop Table*. Click *X* to close the window.
8. Right-click the target object *DMINVA* and click *Target Transformations*.

   The Transformations window opens.
9. Click the *Automap* button.
10. Click the *Validates* tab, and then click the *Insert Intermediate Transforms* button.

    The Validate Calculator opens with the Relational Expression tab selected.
11. Double-click *PRODTYPE* from the Columns/Variables tab.
12. Enter *'Analog'*, with the quotes, under Value, to complete the calculation, and click *OK*.

When you have completed the validation, the Transformations window should appear as shown in the following image:



13. Click *OK* to close the Transformations window.
14. Drag a second target object *DMINVD* from the ibisamp application directory into the workspace, to the right of the SQL object.
15. Right-click the target object *DMINVD* and click *Properties*.

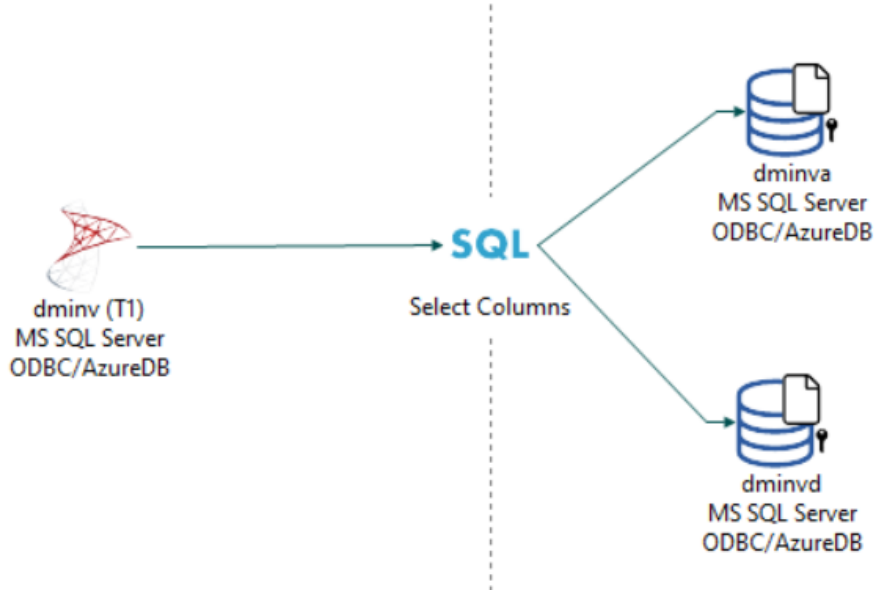
    The Properties dialog box opens.
16. If the flow may be rerun, you should clear the target, using the *Prior to Load Option*. For a relational target, select *Delete all rows from table* or *Truncate Table*. For a FOCUS/FDS target, select *Drop Table*. Click *X* to close the Properties page.
17. Right-click the *DMINVD* target object and click *Target Transformations*.

    The Transformations dialog box opens with the Relational Expression tab selected.
18. Click the *Automap* button.
19. Click the *Validates* tab, and then click the *Insert Intermediate Transforms* button.

    The Validate Calculator opens.

20. Double-click *PRODTYPE* from the Columns/Variables tab.
21. Enter *'Digital'*, with the quotes, under Value to complete the calculation and click *OK*.
22. Click *OK* to close the Transformations window.

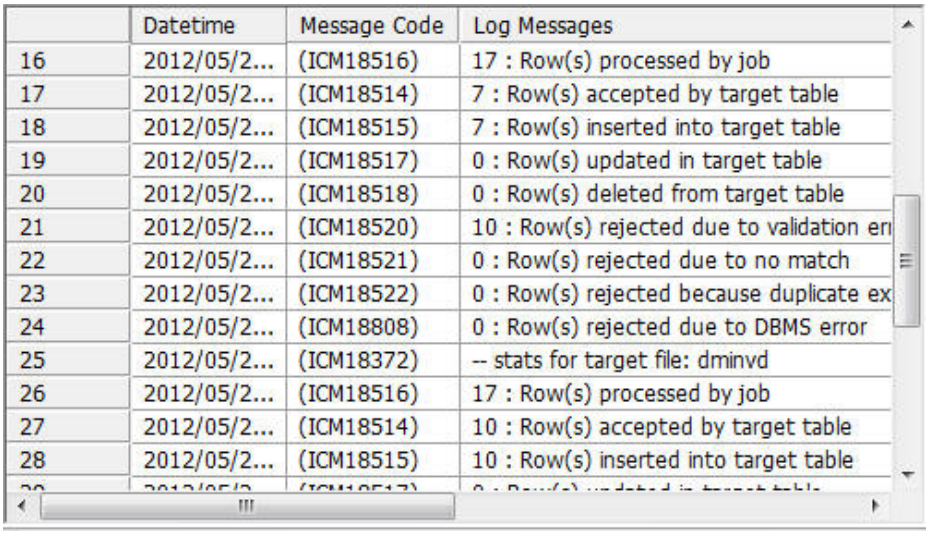
    When you have completed these steps, your data flow should appear as shown in the following image:



23. Save the data flow as DMXSPLIT.
24. On the Flow tab, in the Run group, click *Run* and select *Submit* from the drop-down menu.

    Two data targets are loaded, DMINVA with records containing a PRODTYPE of Analog and DMINVD with records containing a PRODTYPE of Digital.

To view the log, right-click *DMXSPLIT* in the navigation pane, select *Logs*, and then click *Last Log*. The log of the data flow, shown in the following image, indicates that 17 rows were processed, 7 rows loaded into DMINVA, and 10 rows loaded into DMINVD.



	Datetime	Message Code	Log Messages
16	2012/05/2...	(ICM18516)	17 : Row(s) processed by job
17	2012/05/2...	(ICM18514)	7 : Row(s) accepted by target table
18	2012/05/2...	(ICM18515)	7 : Row(s) inserted into target table
19	2012/05/2...	(ICM18517)	0 : Row(s) updated in target table
20	2012/05/2...	(ICM18518)	0 : Row(s) deleted from target table
21	2012/05/2...	(ICM18520)	10 : Row(s) rejected due to validation er
22	2012/05/2...	(ICM18521)	0 : Row(s) rejected due to no match
23	2012/05/2...	(ICM18522)	0 : Row(s) rejected because duplicate ex
24	2012/05/2...	(ICM18808)	0 : Row(s) rejected due to DBMS error
25	2012/05/2...	(ICM18372)	-- stats for target file: dminvd
26	2012/05/2...	(ICM18516)	17 : Row(s) processed by job
27	2012/05/2...	(ICM18514)	10 : Row(s) accepted by target table
28	2012/05/2...	(ICM18515)	10 : Row(s) inserted into target table

Using the Decode Function

DECODE functions are used to convert codes into values. For example, you use DECODE to convert a two-character state code into the complete state name. DECODE can be explicitly stated or you can store the decoded values in a separate file.

The primary use of a DECODE function is to convert codes to full names. To use DECODE in a data flow, build a data flow with transformations that use the function.

In the following example, DMLOCS, a location table, is the data target. Add three transformations to map the source data columns to the target data columns. These transformations are known only to this data flow.

- In the first transformation, STATE, you convert a city abbreviation to its corresponding state code.
- In the second transformation, PLANT_CITY, you turn the city abbreviation into the complete city name.
- In the final transformation, PLANT_STATE, you convert a two-character state code into the full state name, using a list of state codes stored in a separate fixed format file, identified as DMDCD and named DMDCD.FTM.

The location of this state code file must be specified in the data flow so that it can be found at execution time. The procedure DMSTDEF specifies the location of the file, and DMSTDEF should be designated as a pre-extract stored procedure. This means that when the data flow runs, prior to the extraction of data from the source file, the procedure DMSTDEF runs. The file location is now known, and when the PLANT_STATE transformation is calculated, the state file used in the decode is found.

In this example, DMSTDEF contains the following code:

```
FILEDEF DMDCD DISK ibisamp/dmdcd.ftm

-RUN

-TYPE (ICM1) Filedef for State decode file returned &RETCODE
```

This specifies the location of the file used in the DECODE.

Note: If you want to be able to test the DMSTDEF procedure from the DMC, you must first make it available for processing, as illustrated in the example below.

Refer to the sample data flow DMDCODE for the complete example.

Example: Using the DECODE Function in a Data Flow

In the DMC:

1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the right pane, with the SQL object displayed.

   In order to test the query from Data Migrator, you must first specify the location of the DMDCD file. Do this by adding the procedure DMSTDEF to your profile to define its location, and test it before you begin to construct the flow.

2. Right-click a server, select *My Console*, and then click *Edit My profile*.
3. When the Edit User Profile editor opens, enter the line below at the end of the profile:

   EX DMSTDEF

   Click *X* to close the editor, then click *Yes* to save the changes.

4. Drag the data source object *DMORD* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
5. Right-click the SQL object and click *Column Selection*.

   The Column Selection dialog box opens.

6. Select *PLANT* from the Available Columns list, and click the arrow to move it to the Selected Columns list.

7. Click the *Distinct* check box, and click *OK* to close the Column Selection window.
8. Right-click in the workspace to the right of SQL object, select *Add Target*, and then click *Existing*.
9. Select *DMLOCS* from the ibisamp application directory, and click *Select*.

   The data target object DMLOCS is added to the workspace.

10. Right-click the target object *DMLOCS*, and click *Properties*.

    The Properties page opens.

11. If the flow may be rerun, you should clear the target using the *Prior to Load Option*. Since DMLOCS is a fixed format file, select *Delete File*. Click *X* to close.
12. Right-click the *DMLOCS* target object and click *Target Transformations*.

    The Transformations dialog box opens.

13. Click the *Insert Transforms* button.

    The Transformations Calculator opens.

14. Enter *STATE* in the Column field and A2 in the Format field.
15. In the expression field, type

    ```
    DECODE PLANT (BOS MA DAL TX LA CA ORL FL SEA WA STL MO)
    ```

    and click *OK*.

16. Double-click *PLANT_CITY* in the Target Columns list to open the Transformations Calculator.
17. Enter the following as the transformation expression:

    ```
    DECODE PLANT (BOS 'Boston' DAL 'Dallas' LA 'Los Angeles' ORL 'Orlando'
    SEA 'Seattle' STL 'St. Louis')
    ```

18. Click *OK* to close the calculator.
19. Double-click *PLANT_STATE* in the target columns list. This moves it to the Expressions list.
20. Enter the following as the transformation expression:

    ```
    DECODE STATE (DMDCD)
    ```

    Click *OK* to close the calculator.

    When you have defined the three transformations, the Transformations dialog box should appear.

21. Click the *Test transforms* button to test the transformations you just built. Click *Close*.
22. Click *OK* to close the Transformations window.

    Note: If you leave EX DMSTDEF in your profile, then steps 22 to 26 are not needed.

If another user were going to use this flow, then you would need to perform the following steps.

23. Click the *Process Flow* tab.
24. Right-click the arrow between the Start object and the Data Flow object, and click *Delete*. You will be inserting the procedure between them.
25. Drag the procedure, *DMSTDEF*, from the ibisamp application directory in the navigation pane into the process flow workspace.
26. Drag from the *Start* object to the *DMSTDEF* object to connect them, then drag from the *DMSTDEF* object to the *Data Flow* object to connect them.
27. Save the flow as DMXDCODE.
28. On the Flow tab, in the Run group, click *Run*, and then select *Run* from the drop-down menu.

To verify that the target was loaded properly and that the transformations evaluated properly, click the *Data Flow* tab, right-click the *DMLOCS* data target, select *Operations*, and then click *Sample Data*.

Generating a Surrogate Key

Many data warehousing applications make use of a surrogate key instead of using the keys from an operational system, like an alphanumeric employee ID or an integer employee number.

A surrogate key is usually an incrementing integer with no intrinsic meaning. One reason applications use surrogate keys is to improve the performance of JOINs.

You can build a data flow to load a table that has surrogate keys by using a self-referential transformation to generate the values for the surrogate key field.

In the following example, DMSALE, a sales table, is the data source, and the extracted records are loaded into a table named DMREPS. Values for the surrogate key REPNO are generated when the data flow is run.

Refer to the sample data flow DMSURRG for the complete example.

Example: Generating a Surrogate Key in a Data Flow

In the DMC:

1. Right-click an application directory in the navigation pane, point to *New*, and then click *Flow*. The Data Flow tab opens in the right pane, with the SQL object displayed.
2. Drag the data source object *DMSALE* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.

3. Right-click the SQL object and click *Column Selection*.

   The Column Selection dialog box opens.

4. Select *EMPID*, *PLANT*, and *SALESREP* in the Available Columns list and click the arrow to add them to the Selected Columns list.

5. With all three columns selected, select *Group By* from the Aggregate drop-down menu.

6. Click the *Insert Columns* button.

   The SQL Calculator opens.

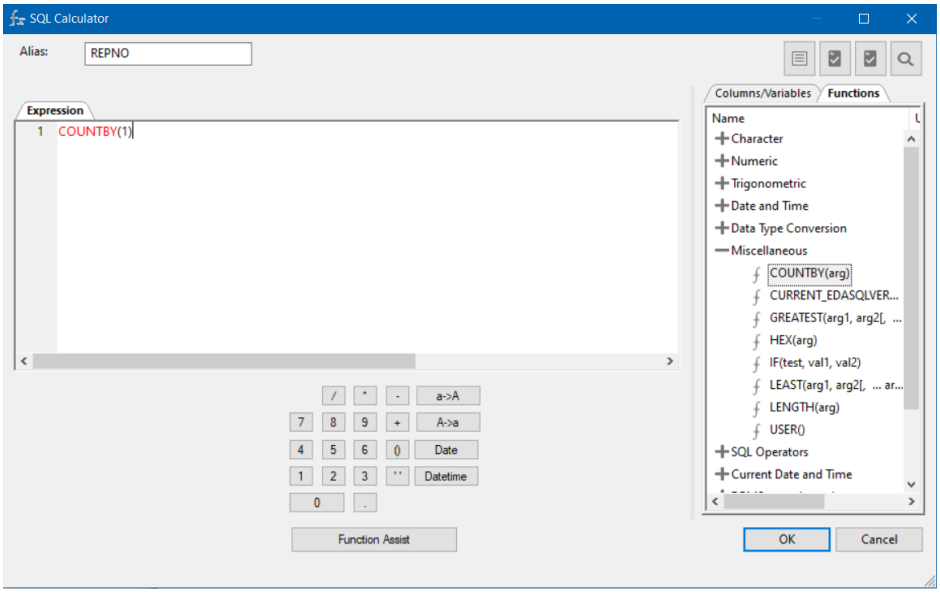
7. In the Alias box, enter *REPNO* for the virtual column.

8. Click the *Functions* tab, expand the *Miscellaneous* folder, and double-click the *COUNTBY* function.

   The Function Assist dialog box opens.

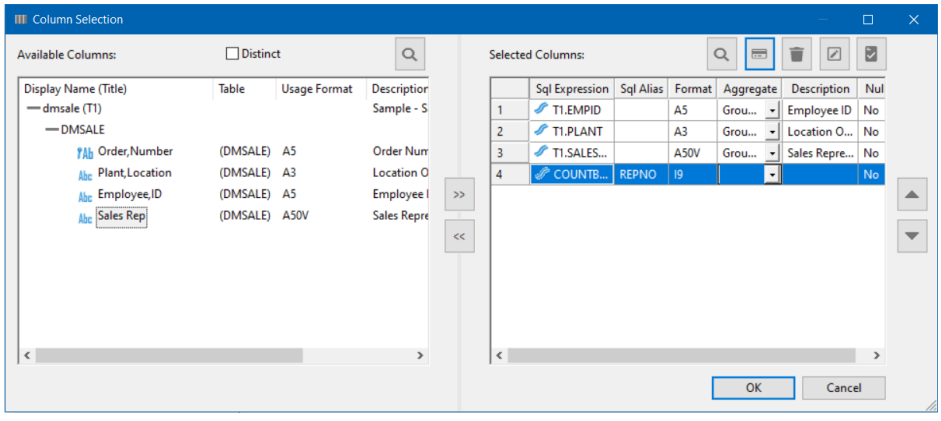
9. Leave the attribute value set to 1 and click *OK*.

   The SQL Calculator should appear as shown in the following image.



10. Click *OK* to close the SQL Calculator.

The Column Selection dialog box should appear as the following image.



11. Click *OK* to close the Column Selector dialog box.

12. Drag the data target object *DMREPS* from the ibisamp application directory into the workspace, to the right of the SQL object.

13. Right-click the target object *DMREPS*, and click *Properties*.

    The Properties dialog box opens.

14. If the flow may be rerun, you should clear the target using the Prior to Load Option drop-down menu. For a relational target, select *Delete all rows from table* or *Truncate Table*. For a FOCUS/FDS target, select *Drop Table*. Close the Properties dialog box.

15. Right-click the target object and click *Target Transformations*.

    The Transformations dialog box opens.

16. Click the *AutoMap* button to map REPNO, EMPID, and PLANT. This moves them to the Expressions list.

    To map *FNAME* and *LNAME*, use substring to extract the necessary characters from the SALESREP source column, of which the first 16 characters are first name and the next 34 characters are last name.

17. Double-click *FNAME* in the Target Columns list.

    The Transformation Calculator opens.

18. Click the *Functions* tab. Expand the *Character* folder and double-click *SUBSTRING*.

    In the Function Assist dialog box, select *SALESREP* from the string drop-down menu.

    Leave the position set as 1 and enter *16* in the length box. Click *OK*.

    The expression now reads:

```
SUBSTRING(SALESREP, 1, 16)
```

19. Click *OK* to close the Transformation Calculator.
20. Double-click *LNAME* in the Target Columns list.

    The Transformation Calculator opens.
21. Click the *Functions* tab. Expand the *Character* folder and double-click *SUBSTRING*.

    In the Function Assist dialog box, select *SALESREP* from the string drop-down menu.

    Enter *17* in the position box and enter *34* in the length box. Click *OK*.

    The expression now reads:

    ```
    SUBSTRING(SALESREP, 17, 34)
    ```
22. Click *OK* to close the Transformation Calculator.
23. To map STATE, double-click *STATE* in the Target Columns list.

    The Transformation Calculator opens.
24. Type the following expression under the Expression tab:

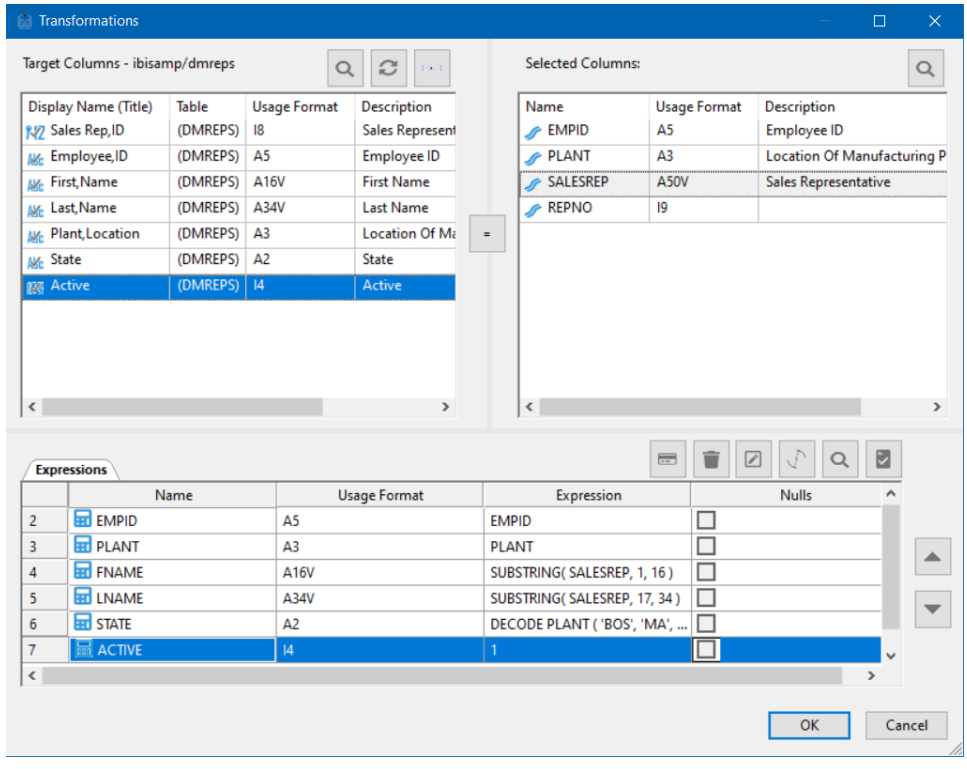
    ```
    DECODE PLANT (BOS MA DAL TX LA CA ORL FL SEA WA STL MO)
    ```

    This expression will convert a three-character city name to its two-character state code.

    Click *OK*.
25. Double-click *ACTIVE* in the Target Column list.

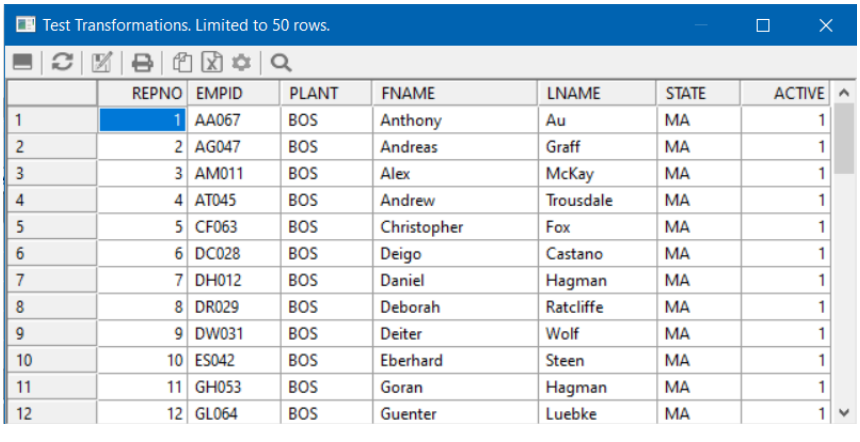
    The Transformation Calculator opens.
26. Type *1* in the Expression tab.
27. Click *OK*.

When you have completed the transformations, the Transformations dialog box should appear, as shown in the following image.



28. Test the transformations by clicking the *Test Transforms* button.

The Test Transformations dialog box opens, as shown in the following image.

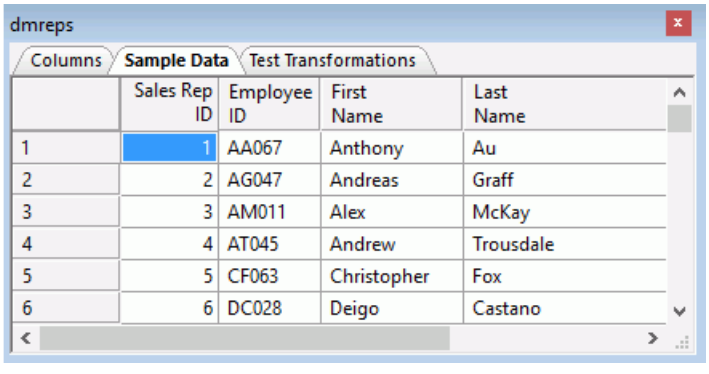


Note that REPNO is incremented by one for each record and that FNAME and LNAME are transformed correctly.

Close the Test Transformations dialog box.

29. Click *OK* to close the Transformations dialog box.
30. Save the data flow as DMXSURRG.
31. On the Flow tab, in the Run group, click *Run*, and then click *Submit* from the drop-down menu.

When the data flow run is complete, verify that the target was loaded properly and that the surrogate key was generated by right-clicking the *DMREPS* object, clicking *Expand*, and then clicking the *Sample Data* tab in the window that opens. The following image shows the DMREPS window with the Sample Data tab selected.



	Sales Rep ID	Employee ID	First Name	Last Name
1	1	AA067	Anthony	Au
2	2	AG047	Andreas	Graff
3	3	AM011	Alex	McKay
4	4	AT045	Andrew	Trousdale
5	5	CF063	Christopher	Fox
6	6	DC028	Deigo	Castano

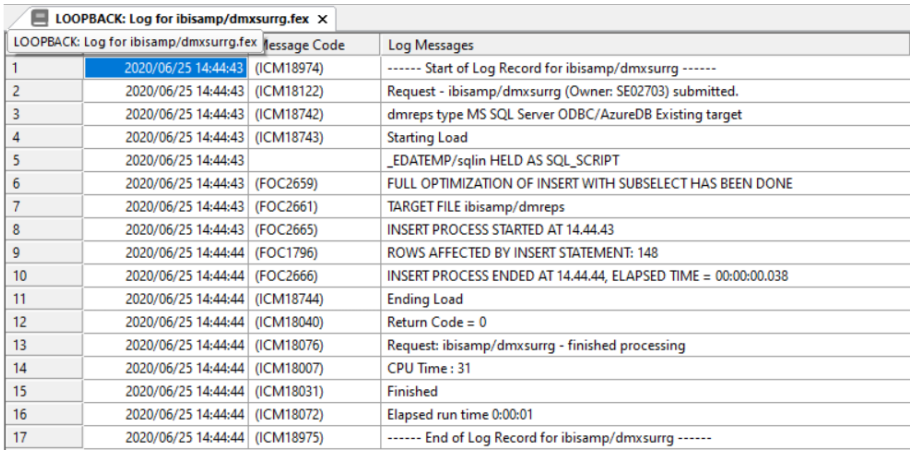
There are two ways to confirm that the job ran correctly:

On the Flow tab, in the Reports group, click *View Last Log*.

or

Right-click the flow from the navigation pane, point to *Logs*, and then click *Last Log*.

The following image shows the log of the most recently run procedure.



LOOPBACK: Log for ibisamp/dmxsurrg.fex

		Message Code	Log Messages
1	2020/06/25 14:44:43	(ICM18974)	------ Start of Log Record for ibisamp/dmxsurrg ------
2	2020/06/25 14:44:43	(ICM18122)	Request - ibisamp/dmxsurrg (Owner: SE02703) submitted.
3	2020/06/25 14:44:43	(ICM18742)	dmreps type MS SQL Server ODBC/AzureDB Existing target
4	2020/06/25 14:44:43	(ICM18743)	Starting Load
5	2020/06/25 14:44:43		_EDATEMP/sqlin HELD AS SQL_SCRIPT
6	2020/06/25 14:44:43	(FOC2659)	FULL OPTIMIZATION OF INSERT WITH SUBSELECT HAS BEEN DONE
7	2020/06/25 14:44:43	(FOC2661)	TARGET FILE ibisamp/dmreps
8	2020/06/25 14:44:43	(FOC2665)	INSERT PROCESS STARTED AT 14.44.43
9	2020/06/25 14:44:44	(FOC1796)	ROWS AFFECTED BY INSERT STATEMENT: 148
10	2020/06/25 14:44:44	(FOC2666)	INSERT PROCESS ENDED AT 14.44.44, ELAPSED TIME = 00:00:00.038
11	2020/06/25 14:44:44	(ICM18744)	Ending Load
12	2020/06/25 14:44:44	(ICM18040)	Return Code = 0
13	2020/06/25 14:44:44	(ICM18076)	Request: ibisamp/dmxsurrg - finished processing
14	2020/06/25 14:44:44	(ICM18007)	CPU Time : 31
15	2020/06/25 14:44:44	(ICM18031)	Finished
16	2020/06/25 14:44:44	(ICM18072)	Elapsed run time 0:00:01
17	2020/06/25 14:44:44	(ICM18975)	------ End of Log Record for ibisamp/dmxsurrg ------

Generating a Surrogate Key Using Slowly Changing Dimensions

Data Migrator includes support for automatically generating surrogate keys when used in a dimension table. This support is only available for relational databases and cannot be used with FOCUS or fixed format file targets.

For details about Slowly Changing Dimension support, see *Target Properties Pane for Relational Targets* on page 290.

Refer to the sample data flow DMSURSCD for the complete example.

Example: Generating a Surrogate Key Using Slowly Changing Dimensions in a Data Flow

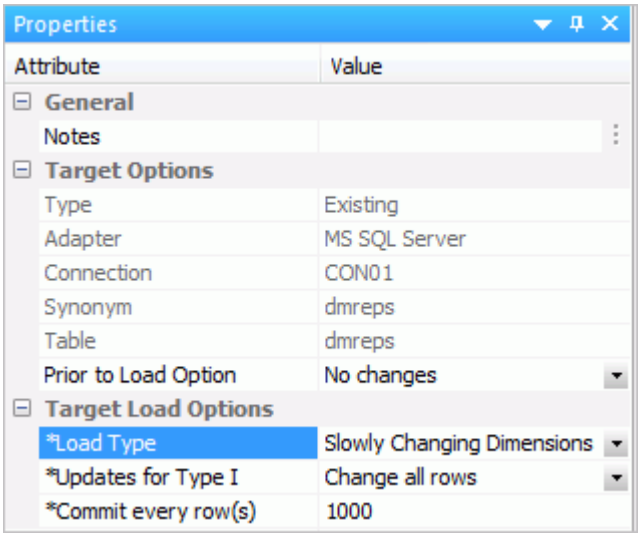
In the DMC:

1. On the Home tab, in the Tools group, click *Options*, and then select *Column Management* in the Options dialog box.
2. Expand the *Business View Editor* folder and select *SCD Type* to display it on the column grid. Click *OK*.
3. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the right pane, with the SQL object displayed.
4. Drag the data source object *DMSALE* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
5. Right-click the SQL object and click *Column Selection*.

   The Column Selection dialog box opens.

6. Select *EMPID*, *PLANT*, and *SALESREP* in the Available Columns list and click the arrow to add them to the Selected Columns list. Click *OK*.
7. Drag the data target object *DMREPS* from the ibisamp application directory into the workspace, to the right of the SQL object.
8. Right-click the target object *DMREPS* and click *Properties*.

   The Properties dialog box opens.
9. If the DMSURRG flow has been run previously, clear the target using the *Prior to Load Option* by selecting *Delete all rows from table* or *Truncate Table*.
10. To enable Slowly Changing Dimension support, for *Load Options* select *Slowly Changing Dimensions*, as shown in the following image.



11. Right-click the target object and click *Target Transformations*.

    The Transformations dialog box opens.
12. Click the *Automap* button to map *EMPID* and *PLANT*. This moves them to the Expressions list.

    To map *FNAME* and *LNAME*, use substring to extract the necessary characters from the SALESREP source column, of which the first 16 characters are first name and the next 34 characters are last name.
13. Double-click *FNAME* and *LNAME* in the target columns list. This moves the columns to the Expressions list.
14. Double-click the *Transform Expression* field for FNAME and type the following expression:

```
SUBSTR (50, SALESREP, 1, 16, 16, 'A16')
```

Click *OK*.

15. Double-click the *Transform Expression* field for LNAME and type the following expression:

```
SUBSTR (50, SALESREP, 17, 50, 34, 'A34')
```

Note: While the creation of a surrogate key does not require the use of the SUBSTR function, it is a useful routine that can be used in a number of different situations.

Click *OK*.

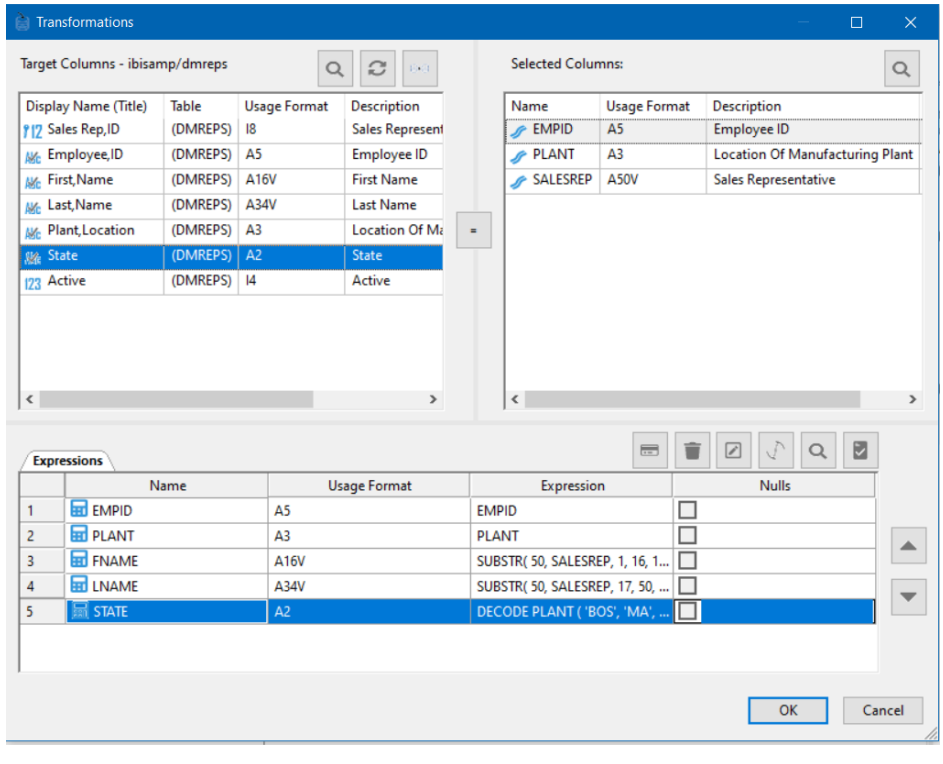
16. To map STATE, double-click *STATE* in the Target Columns list. This moves the column to the Expressions list.

17. Double-click the *Transform Expression* field for STATE and type the following:

```
DECODE PLANT (BOS MA DAL TX LA CA ORL FL SEA WA STL MO)
```

This converts a three character city name to its two character state code.

When you have completed the transformations, the Transformations dialog box should appear, as shown in the following image.

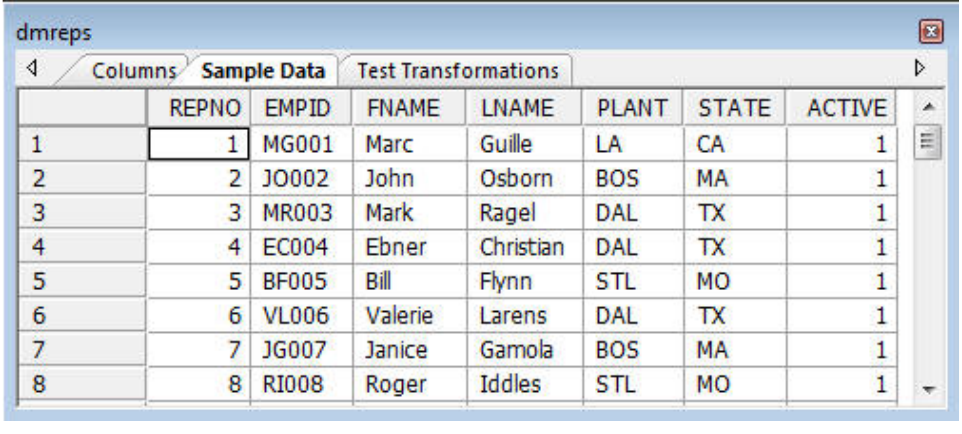


The target table DMREPS has been created with slowly changing dimension attributes assigned for each of its columns. Since REPNO is a key column and the type is blank, it becomes the surrogate key and no transformation is required. The values for ACTIVE are also automatically generated when the data flow is run.

18. Test the transformations by clicking the *Test transforms* button. Click *Close*.
19. Click *OK* to close the Transformations dialog box.
20. Save the data flow as DMXSURSCD.
21. In the Flow tab, in the Run group, click *Run*, and then select *Run* from the drop-down menu.

When the data flow run is complete, verify that the target was loaded properly and that the surrogate key was generated by right-clicking the *DMREPS* object and selecting *Expand*, and then *Sample Data*.

Note that the REPNO surrogate key values are assigned sequentially. However since PLANT and STATE are defined as SCD Type II, an employee is assigned a new surrogate key if the plant they are associated with changes. The ACTIVE flag is set to 1 for their current plant. The following image shows the DMREPS with the Sample Data tab selected.



	REPNO	EMPID	FNAME	LNAME	PLANT	STATE	ACTIVE
1	1	MG001	Marc	Guille	LA	CA	1
2	2	JO002	John	Osborn	BOS	MA	1
3	3	MR003	Mark	Ragel	DAL	TX	1
4	4	EC004	Ebner	Christian	DAL	TX	1
5	5	BF005	Bill	Flynn	STL	MO	1
6	6	VL006	Valerie	Larens	DAL	TX	1
7	7	JG007	Janice	Gamola	BOS	MA	1
8	8	RI008	Roger	Iddles	STL	MO	1

There are two ways to confirm that the job ran correctly:

On the Flow tab, in the Reports group, click *View Last Log*.

or

Right-click the flow from the navigation pane, select *Logs*, and then click *Last Log*.

Mapping to a Sequential File of a Specific Format

Situations arise where you need to extract data from data sources and load a sequential table with a specific layout. For example, you may need to load a packaged application of a very specific format with a mix of both packed and alphanumeric columns. The data you load must be in the order the packaged application expects and the formats must match exactly. This means that when you build your data flow, you must make sure that each column is in the expected order and format.

To do this, build a data flow to load a sequential file with a previously defined synonym. This means that you create the synonym to match the format required by the application you are loading and you use that synonym as the target of the data flow.

In the following example, DMHR, a human resources table, is the data source, and extracted records are loaded into a data target sequential file named DMHRINP. A synonym describing the layout of this sequential table was created in the format of the sequential file that is being loaded. The sequential file includes two packed columns and two alphanumeric columns.

Refer to the sample data flow DMSEQ for the complete example.

Example: Mapping to a Sequential File in a Data Flow

In the DMC:

1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the right pane, with the SQL object displayed.
2. Drag the data source object *DMHR* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Right-click the SQL object and click *Column Selection*.

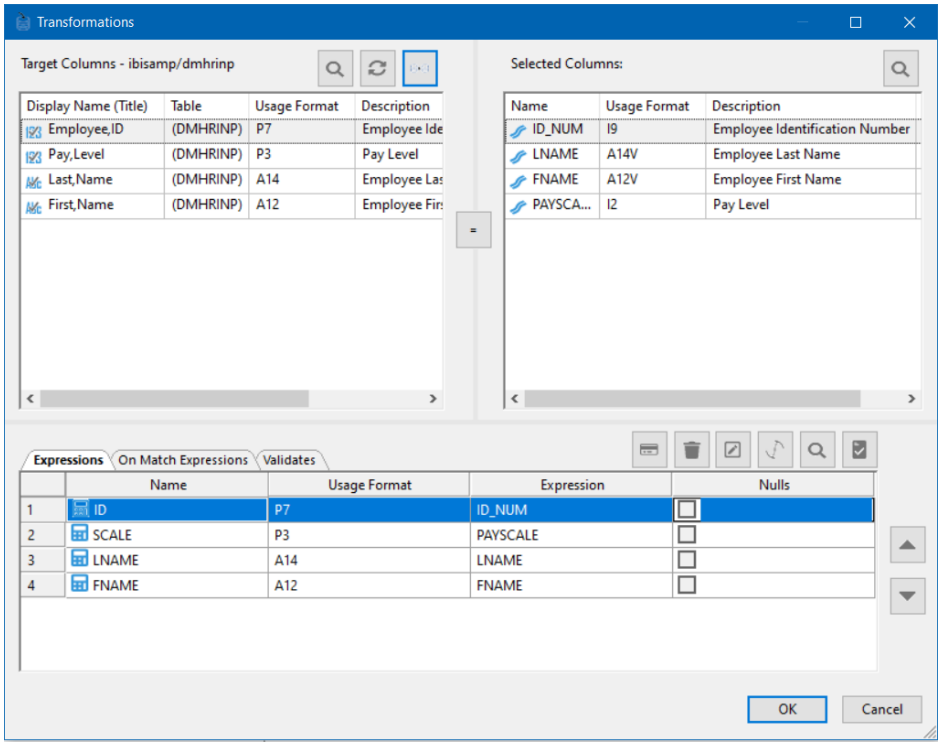
   The Column Selection window opens.
4. Select *Employee,ID (ID_NUM)*, *Last,Name (LNAME)*, *First,Name (FNAME)*, and *Pay,Level (PAYSCALE)* in the Available Columns list and click the arrow to add them into the Selected Columns list. Click *OK*.
5. Drag the data target object *DMHRINP* from the ibisamp application directory into the workspace, to the right of the SQL object.
6. Right-click the target object *DMHRINP* and click *Properties*.

   The Properties dialog box opens.
7. If the flow may be rerun, you should clear the target using the *Prior to load* option. Since dmhrinp is a fixed format file, select *Delete File*. Close the Properties dialog box.
8. Right-click the target object and click *Target Transformations*.

   The Transformation window opens.

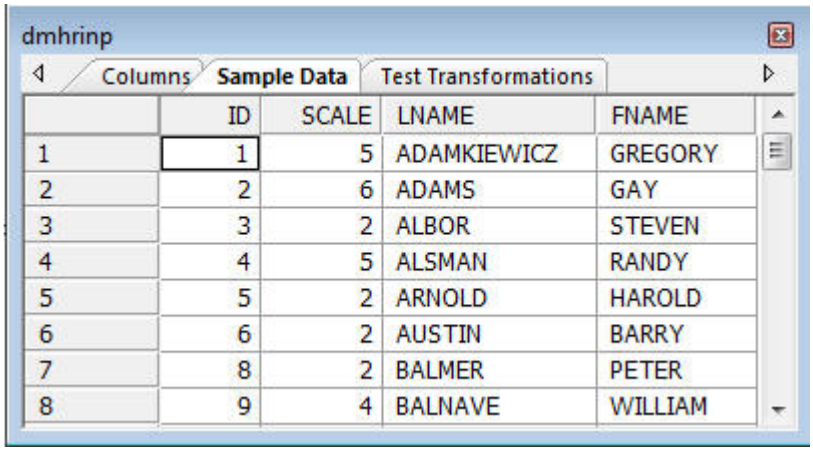
9. Select *Employee,ID (ID)* in the Target Columns list and *ID_NUM* in the Selected Columns list, then click the equal sign (=). The relation is added to the Expressions list.
10. Select *Pay,Level (SCALE)* in the Target Columns list and *PAYSCALE* in the Selected Columns list, then click the equal sign (=). The relation is added to the Expressions list.
11. Click the *Automap* button to map *LNAME* and *FNAME*.

    When you have completed the transformations, the Transformations window should appear, as shown in the following image:



12. Test the transformations by clicking the *Test transforms* button. Click *Close*.
13. Click *OK* to close the Transformations window.
14. Save the data flow as DMXSEQ.
15. On the Flow tab, in the Run group, click *Run*, and then select *Run* from the drop-down menu.

When the data flow run is complete, verify that the data was loaded properly in the sequential file by right-clicking the *DMHRINP* object, selecting *Expand*, and clicking the *Sample Data* tab, as shown in the following image.



	ID	SCALE	LNAME	FNAME
1	1	5	ADAMKIEWICZ	GREGORY
2	2	6	ADAMS	GAY
3	3	2	ALBOR	STEVEN
4	4	5	ALSMAN	RANDY
5	5	2	ARNOLD	HAROLD
6	6	2	AUSTIN	BARRY
7	8	2	BALMER	PETER
8	9	4	BALNAVE	WILLIAM

Creating and Using Reusable Transformations

There are many situations where you need to perform the same calculation on several different columns. It could be something as simple as converting a date to a different format, adding dashes to an id number so that it displays as you would expect a social security number to display, or something much more complex that requires you to perform the same calculations on different columns of data. Instead of coding the calculation over and over for each column, it would be more efficient to create the calculation once, and then use it whenever you need that calculation. In essence, you create a reusable transformation.

Before you build the transformation, you must first create the function to perform the necessary transformations. Functions are created in the DMC as a server object with the type Define Function.

This technique is divided into two parts. In the first part, we build the function. In the second, we use the function in transformations within a data flow, then incorporate the data flow object into a process flow to make it available at execution time. In the following example, we build two functions:

- **DMPROPER** converts a name to last name, first name format, where the first letter of each name is in upper case and the rest of the name is in lowercase.
- **DMDATFMT** converts a date to 2009-JUL-04 format, which includes dashes between the year, month, and day.

These routines are stored in a stored procedure which is identified in the Data Management Console with a distinguished icon and a type of Userin Functions. In the following example, DMHR, a human resource table, is the data source, and the extracted records are loaded into a new data target named EMPDATES. Employee name (PROPERNAME), start date (START), and termination date (TERM) are calculated using the define functions.

Refer to the sample data flows in the ibisamp directory, DMREUSE, for the complete example. The User Functions procedure DMFNS contains the sample function DMPROPER and DMDATFMT.

Example: Creating the DEFINE Functions

The first function, which we call DMPROPER, takes two parameters, a last name and a first name. It returns a formatted proper name.

In the DMC:

1. Right-click an application directory in the navigation pane, point to *New*, and then click *User Functions*.

   The Define Function View opens in the workspace in tree view.

2. Right-click *FUN1* andselect *Properties*. In the Properties panel, change the name to *DMPROPER*.
3. Under the Parameters folder, right-click *PARAM1* and select *Properties*.

   The Properties pane opens.

4. In the PARAM field box, click the *PARAM1* value and type the name of the first parameter *LASTNAME*. In the FORMAT field box, type in the value *A17*.

   Remove *1* from the DEFAULT field box in the Miscellaneous section. Select the *Show Field List* check box.

5. Right-click the *Parameters* folder, point to *Insert*, and then click *Parameter*. In the PARAM field box in the properties panel, click the *PARAM1* value and type the name of the first parameter *FIRSTNAME*. In the FORMAT field box, type in the value *A14*. Under Miscellaneous, select the *Show Field List* check box.
6. Under the Defines folder, click *FUN1*. Its properties panel opens to the right. In the EXPRESSION field box, click the ellipsis  button.

   The Define Calculator opens.

7. Enter *A34V* in the Format field.

   Type the following in the Expression field:

```
INITCAP( LASTNAME ) || (', ' | INITCAP( FIRSTNAME ))
```

8. Click *OK* to close the calculator.
9. Click *Save As* in the Quick Access Toolbar and save the function file as *dmfns*.

The second function converts a date to a common format.

1. Right-click *dmfns*, point to *Insert*, and then click *Function*.

   A new function prototype is added.

2. Right-click *FUN1* andselect *Properties*. In the Properties panel, change the name to *DMDATFMT*.
3. Expand the *Parameters* folder, click *PARAM1*.

   The Properties pane opens.

4. In the PARAM box, click the *PARAM1* value and type the name of the first parameter *INDATE*. In the FORMAT box, type in the value *YYMD* and in the DEFAULT box under the Miscellaneous section, type in the value *&YYMD*. Select the *Show field list* check box.
5. Right-click DMDATFMT in the *Defines* folder, and click *Properties*.

   The Properties pane opens.

6. In the EXPRESSION field box, click the ellipses. The Define Calculator opens.
7. Click the *Functions* tab on the right pane of the Define Calculator to see the list of available functions.

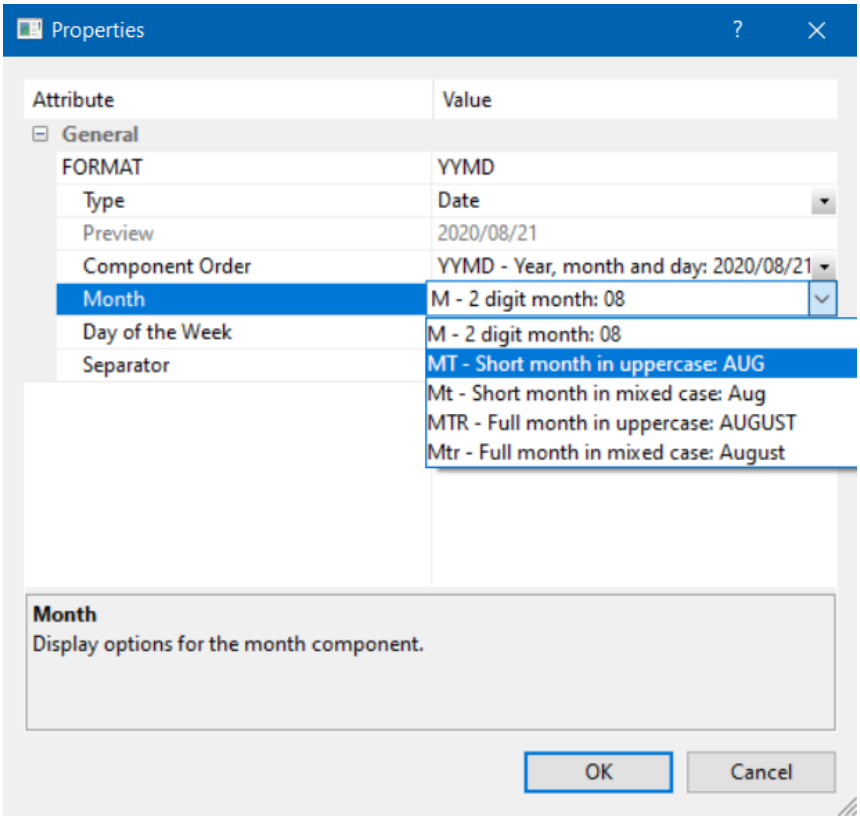
   Note: You can right-click the Name heading bar to toggle the view between List View (lists the functions alphabetically) and Tree View (lists them by category).

8. Expand the *Format Conversion* folder and double-click the *DT_FORMAT* function.

   The Function Assist panel opens.

9. Select *INDATE* from the date drop-down menu.
10. Click the ellipses in the date_format box and select *Date* from the drop-down menu.

11. From the Month drop-down menu, select *MT - Short month in uppercase:AUG*, as shown in the following image.



12. From the Seperator drop-down menu, select *Hyphen*.

13. Click *OK* to close the Properties dialog box.

14. Click *OK* to close the Function Assist dialog box.

15. To verify the DT_FORMAT(INDATE, 'YY-MTD') expression, click the *Check Expression* button on the toolbar of the Define Calculator.

    Click *OK* to close the Define Calculator.

16. To test the transform, right-click *DMDATFMT* and click *Test Run* from the context menu.

    The Test Function window opens, showing the INDATE value in the 'YYMD' format and the DMDATFMT value in the 'YY-MTD' format.

17. Click *Save* from the Quick Access Toolbar.

Note: When creating a define function, the following names must be the same:

- The name of the function.
- The name of the last define in the function.

Example: Using the DEFINE Functions

In the DMC:

1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the workspace, with the SQL object displayed.
2. Drag the data source object *DMHR* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Right-click the *DMHR* object and click *Source Transformations*.

   The Source Transformations window opens.
4. Click the *Insert Transforms* button.

   The Transformations Calculator opens.
5. Enter *PROPERNAME* in the Name field and *A30* in the Format field.
6. Click the *Functions* tab and expand the *User Defined* folder, followed by the *baseapp* directory and the *dmfns* procedure.
7. Double-click *DMPROPER*.

   The *Function Assist* window opens. Select *LNAME* and *FNAME* from the Field list drop-down menus for the parameters.
8. Click *OK* to close the Function Assist window and the Transformation Calculator.
9. Click the *Insert transforms* button.
10. Enter START in the Column field and A12 in the Format field.

    Click the *Functions* tab. Expand the *User Defined* folder, the *baseapp* directory, and the *dmfns* procedure.
11. Double-click the *DMDATFMT* function. The function assistant opens. Select START_DATE as the parameter for the function.

    Select *ON* from the Missing drop-down list.

    Click *OK* to close the Function Assist and the Transformations Calculator.
12. Click the *Insert Transforms* button. Enter *ENDDATE* in the Column field and *A12* in the Format field.

13. Click the ellipsis  button in the Format field.

    Select *ON* from the Missing drop-down list.

    Repeat the above procedure to select the DMDTEFMT function, using TERM_DATE as the parameter.

14. When you have completed the transformations, the Source Transformations window opens.
15. Click *OK* to close the Transformations window.
16. Right-click the SQL object and click *Column Selection*.

    The Column Selection window opens.

17. In the Available Columns list, select *PROPERNAME*, *START*, and *ENDDATE* and click the arrow to add them to the Selected Columns list.
18. To test the transformations, click the *Test SQL Statement* button. Note that the transformations are correct, with PROPERNAME correctly defined, and the correct START and ENDDATE dates.
19. Click *X* to close the Test SQL window, and *OK* to close the Column Selection window.
20. Right-click the workspace to the right of the SQL object, select *Add Target*, and then click *New*.
21. In the new target, right-click and click *Properties*.
22. In the Properties panel, select the Adapter you used for your sample procedures, the application directory you are using for your samples, and enter *EMPDATES* as the synonym. Click *X* to close the panel.
23. Save the flow as DMXREUSE.
24. On the Flow tab, in the Run group, click *Run*, and then select *Submit* from the drop-down menu.

To verify that the transformations have worked properly, click the *Data Flow* tab and right-click the *EMPDATES* data target and select *Expand*. Click the *Sample Data* tab to review the output.

Using Multiple Targets to Create a Single File With Multiple Records Types

This flow uses two targets that both point to the same fixed format file to write out two different record types from the HR file. There is a start record for all employees, but there is only a termination record for employees that have been terminated which is accomplished with a validate. The two different record types are designated by assigning a value to the RECTYPE field in the targets.

Refer to the sample data flow DMRCTYP for the complete example.

In the DMC:

1. Right-click an application directory, select *New*, and then click *Flow*. The Data Flow tab opens in the workspace, with the SQL object displayed.
2. Drag the *DMHR* source from the ibisamp application directory to the left of the SQL object.
3. Right-click the SQL object and click *Column Selection*.
4. Select *Employee,ID (ID_NUM)*, *Starting,Date (START_DATE)*, *Termination,Date (TERM_DATE)*, *Current,Status (STATUS)*, *POSITION*, and *Pay,Level (PAYSCALE)* in the available columns list, and click the arrow to add them to the Selected columns list.
5. Click *OK*.
6. In the ibisamp application directory, drag *DMSTART* and *DMTERM* to the right of the SQL object.
7. Double-click *DMTERM* to open the Transformations window. Click the *Automap* button.
8. Click the *Insert Transforms* button. The Transformation Calculator opens. For the name type RECTYPE, for the format type A1, and for expression type 'T'.
9. Double-click *SEVERENCE* to add it to the Expressions. For the expression, enter PAYSCALE * 500.
10. Select the Validates tab and click *Insert Transforms*. The Transformation Calculator opens.
11. Select the Expression tab and enter TERM_DATE NE MISSING. Click *OK*.
12. Click *OK*.
13. Double-click *DMSTART* to open the Transformations window. Select the *Automap* button.
14. Select the *Insert Transforms* button. The Transformation Calculator opens. For the name, type RECTYPE. For the format, type A1 and for expression, type 'S'.
15. Click *OK*.
16. On the Flow tab, in the Run group, click *Run*, and then select *Submit* from the drop-down menu.

17. On the Flow tab, in the Reports group, click *View Last Log*. Note that 223 Start records and 41 Termination records were loaded.

Message Code	Log Messages
(ICM18974)	------ Start of Log Record for ibisamp/dmrctyp ------
(ICM18122)	Request - ibisamp/dmrctyp (Owner: dm) submitted.
(ICM18742)	dmstart type Flat File Existing target
(ICM18742)	dmterm type Flat File Existing target
(ICM18429)	Issuing PREPARE
(ICM18743)	Starting Load
	0 WARNING.. ON MATCH INCLUDE INPUTS DUPLICATE SEGMENTS
	0 WARNING.. ON MATCH INCLUDE INPUTS DUPLICATE SEGMENTS
(FOC1291)	RECORDS AFFECTED DURING CURRENT REQUEST : 264/INSERT
(ICM18041)	-- stats for source file
(ICM18516)	223 : Row(s) processed by job
(ICM18519)	0 : Row(s) rejected due to format error
(ICM18372)	-- stats for target file: dmstart
(ICM18516)	223 : Row(s) processed by job
(ICM18514)	223 : Row(s) accepted by target table
(ICM18515)	223 : Row(s) inserted into target table
(ICM18517)	0 : Row(s) updated in target table
(ICM18518)	0 : Row(s) deleted from target table
(ICM18520)	0 : Row(s) rejected due to validation errors
(ICM18521)	0 : Row(s) rejected due to no match
(ICM18522)	0 : Row(s) rejected because duplicate exist
(ICM18808)	0 : Row(s) rejected due to DBMS error
(ICM18372)	-- stats for target file: dmterm
(ICM18516)	223 : Row(s) processed by job
(ICM18514)	41 : Row(s) accepted by target table
(ICM18515)	41 : Row(s) inserted into target table
(ICM18517)	0 : Row(s) updated in target table
(ICM18518)	0 : Row(s) deleted from target table
(ICM18520)	182 : Row(s) rejected due to validation errors
(ICM18521)	0 : Row(s) rejected due to no match
(ICM18522)	0 : Row(s) rejected because duplicate exist
(ICM18808)	0 : Row(s) rejected due to DBMS error
(ICM18744)	Ending Load
(ICM18040)	Return Code = 0
(ICM18076)	Request: ibisamp/dmrctyp - finished processing
(ICM18007)	CPU Time : 46
(ICM18031)	Finished
(ICM18072)	Elapsed run time 0:00:01
(ICM18975)	------ End of Log Record for ibisamp/dmrctyp ------

18. To see the output with the two different record types in the ibisamp directory with no filter, scroll down to dmrecs - Fixed Format File. Next, right-click *dmrecs* and click *Edit*. Scroll down to see the termination records.

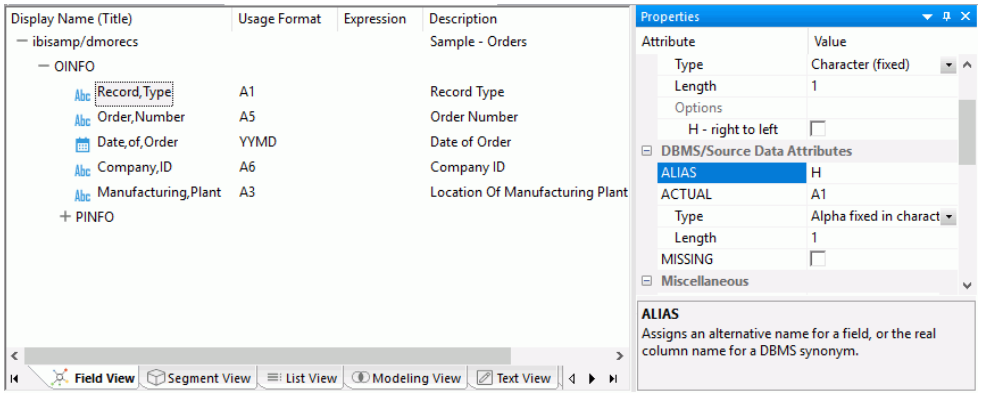
Creating a Target File With Multiple RECTYPEs

Instead of loading a target, you can also use Data Migrator to extract data from a data source and create input files for a packaged application that requires RECTYPEs indicating different record layouts.

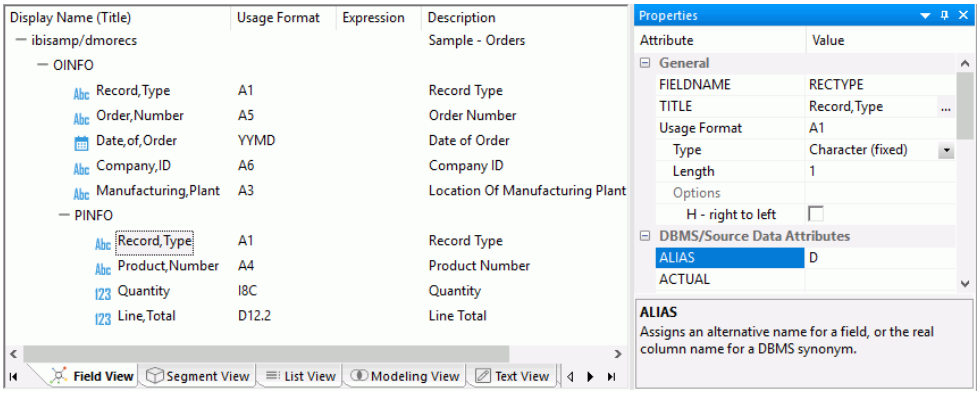
Data Migrator can create a fixed format file containing the different record layouts. This requires a data flow that loads a sequential file with a previously defined layout. The synonym for the sequential file contains one segment for each RECTYPE, along with all the associated columns. The RECTYPE identifies the type of record in the ALIAS. Each segment has a unique value for RECTYPE.

The following example uses a single synonym with multiple segments to create a fixed format file containing two RECTYPEs. DMORD, an orders table, is the data source. Extracted records are loaded into one sequential file data target, DMORECS, which contains two segments.

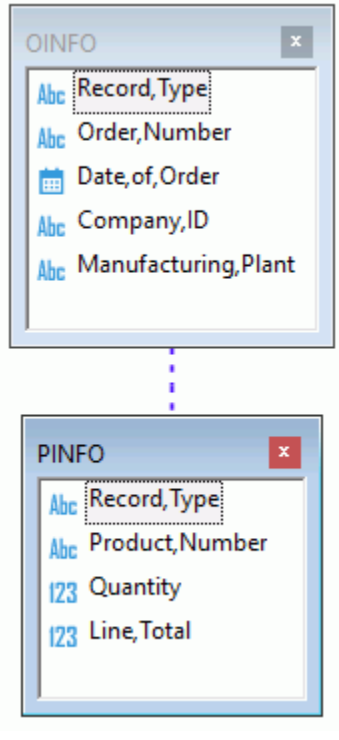
The first segment contains the header information, indicated by a RECTYPE of 'H', and includes the columns ORDER_DATE, STORE_CODE, and PLANT, as shown in the following image.



The second segment contains the detail information, indicated by a RECTYPE of 'D', and includes the columns PROD_NUM, QUANTITY, and LINEPRICE, as shown in the following image.



The graphical view of the DMORECS synonym shows how the two segments are related. OINFO contains the header records and PINFO contains the detail records, as shown in the following image.



Refer to the sample data flow ORECS for the complete example.

Example: Creating a Fixed Format File With Different Record Layouts

In the DMC:

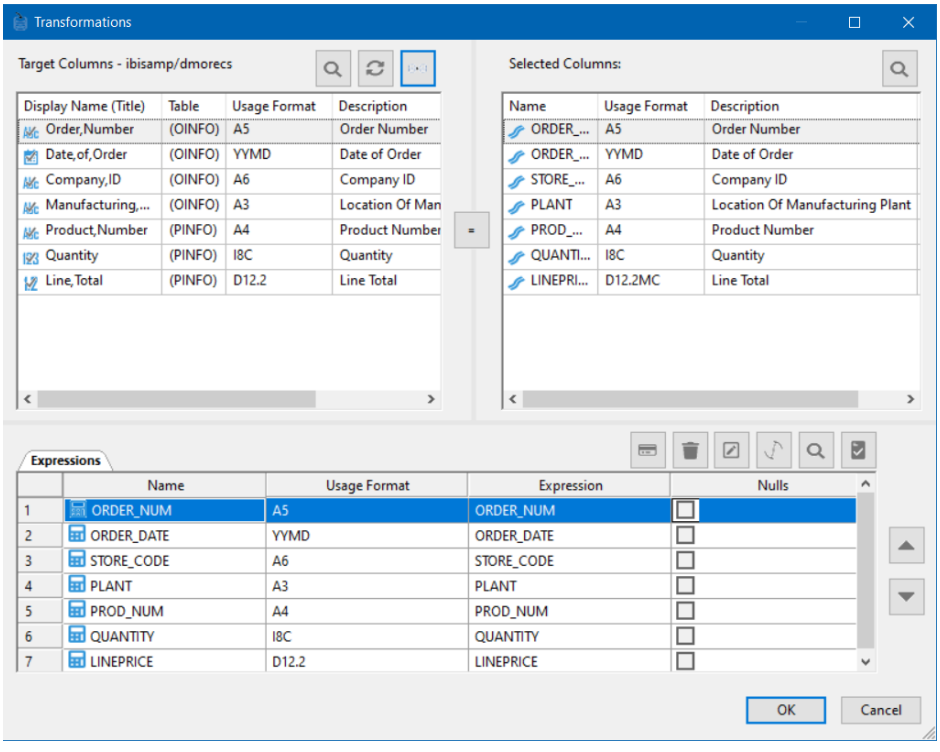
1. Right-click an application directory, select *New*, and then click *Flow*. The Data Flow tab opens in the workspace, with the SQL object displayed.
2. Drag the *DMORD* data source object from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Right-click the *SQL object* and click *Column Selection*.

   The Column Selection dialog box opens.
4. Select *Order,Number (ORDER_NUM)*, *Date of,Order (ORDER_DATE)*, *Company,ID (STORE_CODE)*, *Plant,Location (PLANT)*, *Product,Number (PROD_NUM)*, *QUANTITY*, and *Line,Total (LINEPRICE)* in the Available Columns list and click the arrow to add them to the Selected Columns list.
5. Click *OK*.
6. Drag the *DMORECS* data target object from the ibisamp application directory into the workspace, to the right of the SQL object.
7. Right-click *DMORECS* and click *Properties*.

   The Properties dialog box opens.
8. If the flow will be rerun, you should clear the target using the *Prior to Load Option*. Since DMORECS is a fixed format file, select *Delete File*. Click *X* to close the Properties panel.
9. Right-click *DMORECS* and click *Target Transformations*.

   The Transformation dialog box opens.
10. To map the source columns to the data target, click the *AutoMap* button.

The Transformations should appear as shown in the following image.

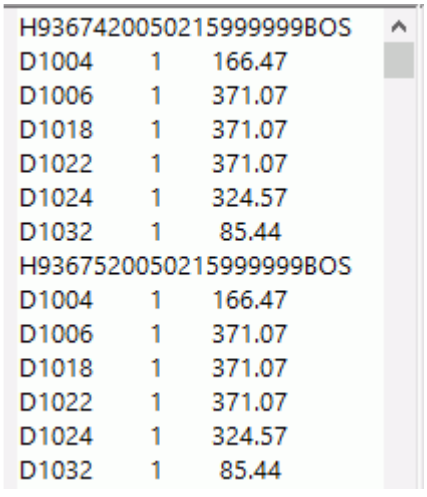


Note that the RECTYPE columns remain unmapped. The RECTYPE values are assigned automatically at load time based on the value given for the RECTYPE in the synonym, in this case, H or D.

11. Click *OK*.
12. Save the data flow as *DMXRCTYP*.
13. On the Flow tab, in the Run group, click *Run* and select *Submit* from the drop-down menu.

To view the output file created by the data, ensure that the No Filter button is selected in the Browser. Expand the *baseapp* directory. There you will see a filename of dmorecs with a type of fixed format file. Double-click it to open it.

Note that the output file contains records with two different layouts, as shown in the following image.



Combining Multiple Source Rows in a Single Target Row

Sometimes data from multiple input rows needs to be combined into a single row. For example, you might encounter a situation where three separate records from a data source map to one record in the data target.

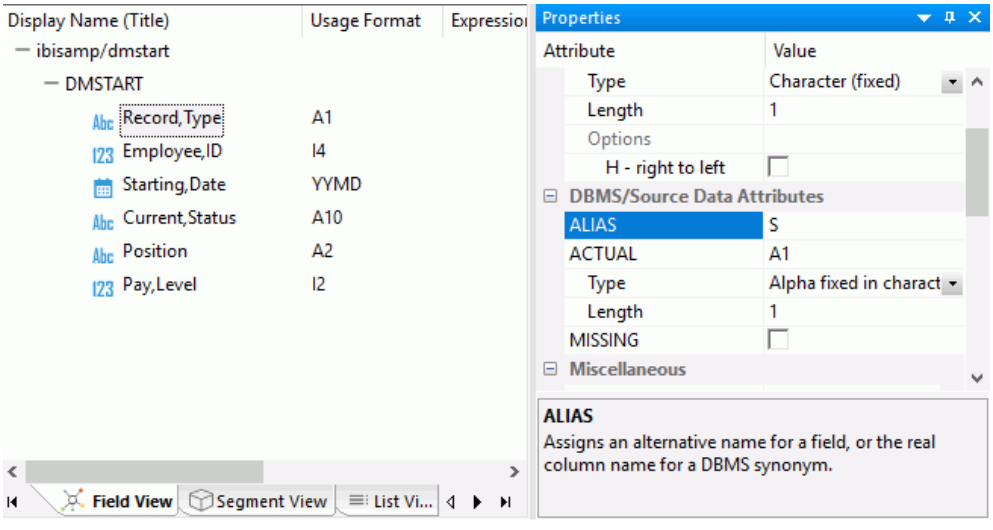
In the following example, the data source contains transaction records with an ID number, action code, and date. The action code, with values of either S or T, indicates whether the date is a Start Date or Termination Date. For example, the data source contains:

```
ID_NUM ACTION  ACT_DATE
------ ------  --------
    23 S       1997/03/26
    24 S       1995/08/24
    24 T       1999/02/16
    25 S       1995/10/05
    25 T       1999/01/15
```

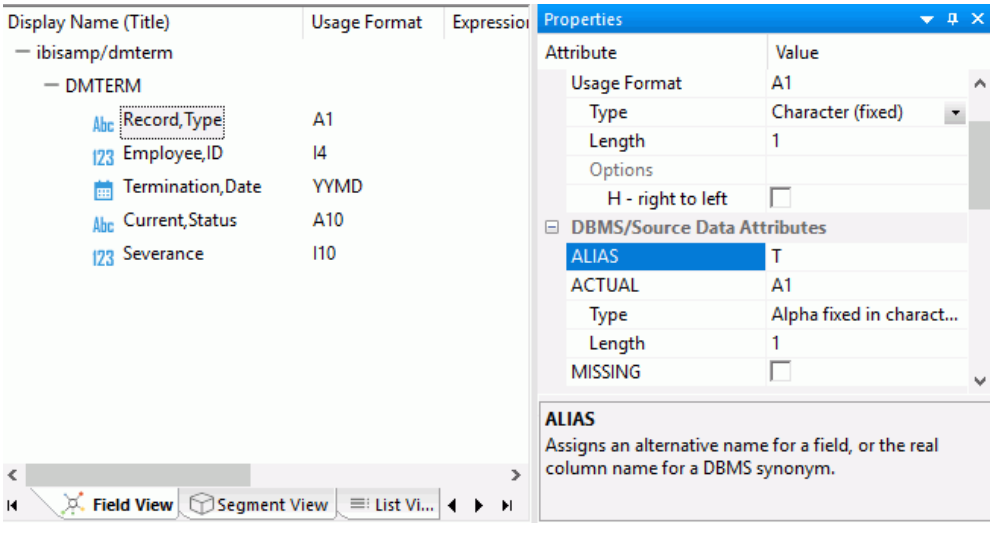
The data flow takes the source data and flips, or pivots, it so that the data target contains one row for each ID_NUM, each with two date columns: one for the Start Date, and one for the Termination Date (if any).

This solution uses two different synonyms to describe the same source file. Each synonym recasts the ACTION column to a RECTYPE, allowing the file to be split into Start Date records and End Date records. A select against DMSTART automatically selects only those records with a RECTYPE of S and a select against DMTERM retrieves only those with a RECTYPE of T.

The synonym for DMSTART shows a RECTYPE of S, which would retrieve only Start Date records, as shown in the following image.



The synonym for DMTERM shows a RECTYPE of T, which would retrieve only End Date records, as shown in the following image.



When the job is run, the output table for the ID_NUM values shown above is:

```
ID_NUM START_DATE  TERM_DATE
------  ----------  ---------
    23 1997/03/26  .
    24 1995/08/24  1999/02/16
    25 1995/10/05  1999/01/15
```

Since there was no T row in the input file for ID_NUM value 23, there is no TERM_DATE value in the output.

Note:

- This example uses the output from the data flow DMRCTYP as a source. Therefore, you must execute DMRCTYP before you execute this request.
- This example can easily be extended to support more than two record types by adding a synonym for each record type.

Refer to the sample data flow DMPIVOT for the complete example.

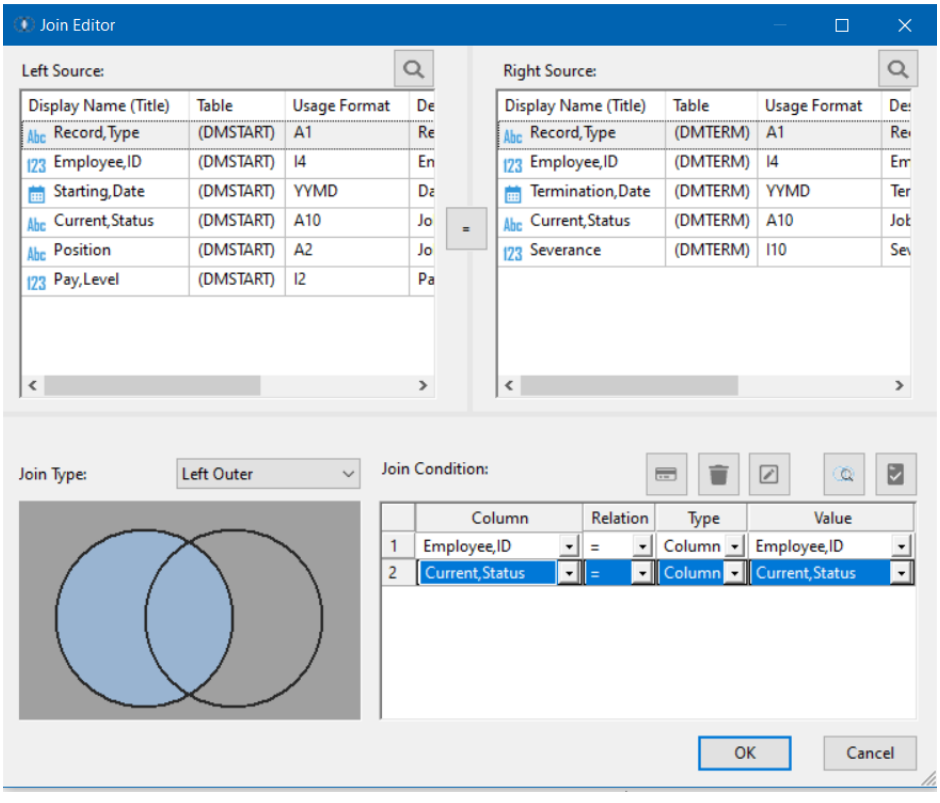
Example: Combining Data Source Rows in a Single Data Target Row

In the DMC:

1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the workspace, with the SQL object displayed.
2. Drag the data source object *DMSTART* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Drag a second data source object, *DMTERM*, into the workspace.
4. To define the Join condition, right-click the Join object and click *Join Editor*.

   The Join Editor window opens.
5. Select *Employee,ID (ID_NUM)* in the Left and Right Source Columns lists and click the equal (=) sign.
6. To make sure that you obtain those records that have a START_DATE but not an END_DATE, the JOIN type should be a left outer Join. Click the left circle in the Join Type diagram.

When you have completed the Join, it should appear, as shown in the following image:



7. Click *OK* to close the Join Editor window.
8. Right-click the SQL object and click *Column Selection*.

   The Column Selection window opens.
9. Select *Employee,ID (ID_NUM)* and *Starting,Date (START_DATE)* from the DMSTART segment, and *Termination,Date (TERM_DATE)* from the DMTERM segment in the Available Columns list and click the arrow to add them into the Selected Columns list.
10. Click *OK*.
11. Right-click the workspace to the right of the SQL object, select *Add Target*, and then click *New*.
12. In the Properties window, select the adapter you used for your sample procedures, the application directory you are using for your samples, and enter *DMXEDATE* as the synonym. Close the Properties box.
13. Save the data flow as DMXPIVOT.

14. Run *ibisamp/dmxpivot*.
15. On the Flow tab, in the Run group, click *Run*, and then select *Run* from the drop-down menu.

When the data flow run is complete, verify that the data was loaded properly by right-clicking the *DMXEDATE* object, selecting *Expand*, and clicking the *Sample Data* tab.

Using Lookups to Create Multi-Table Flows Without Joins

Data Migrator now allows you to build a single data flow that performs lookups into multiple tables using the DB_LOOKUP function without requiring explicit joins.

In the following example, DMORD is the primary source containing orders data, while DMCOMP and DMINV are the lookup sources. While DMORD contains store codes and product numbers, DMCOMP contains the store names for those store codes, and DMINV contains the product names for each product code.

Refer to the sample data flow DMLKUPS for the complete example.

For more information on improving lookup performance, see *Improving Lookup Performance* on page 480.

Example: Creating a Data Flow Using Lookups

This example uses the DB_LOOKUP function to create two target transformations.

1. In the DMC, right-click an application directory in the navigation pane, point to *New*, and then click *Flow*. The Data Flow tab opens in the workspace, with the SQL object displayed.
2. Drag the data source object *DMORD* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Double-click the *SQL* object to open the Column Selection dialog box.
4. Select the *Company,ID (STORE_CODE)* and *Product,Number (PROD_NUM)* columns and click the double arrows button to add them to the Selected Columns section.
5. Select *Group By* from the Aggregate drop-down menu for both columns.
6. Select the *QUANTITY* and *Line,Total (LINEPRICE)* columns and click the double arrows button to add them to the Selected Columns.
7. Select *Sum* from the Aggregate drop-down menu for these two columns.
8. Click *OK* to close the Column Selection dialog box.
9. Right-click the workspace to the right of the SQL object, point to *Add Target*, and then click *Existing*.

The Select Synonym dialog box opens.

10. Select the *ibisamp* directory.

    Select the *dmstores* synonym and click the *Select* button.

11. Double-click the *dmstores* target object or right-click it and click *Target Transformations*.

    The Transformations window opens.

12. Click the Automap button to map the identically named columns.

13. Double-click *STORENAME* to add it to the transformation list, then click the calculator button.

    The Transformation Calculator opens.

14. Click the *Functions* tab, expand the *Data Source and Decoding* folder, and double-click the *DB_LOOKUP* function.

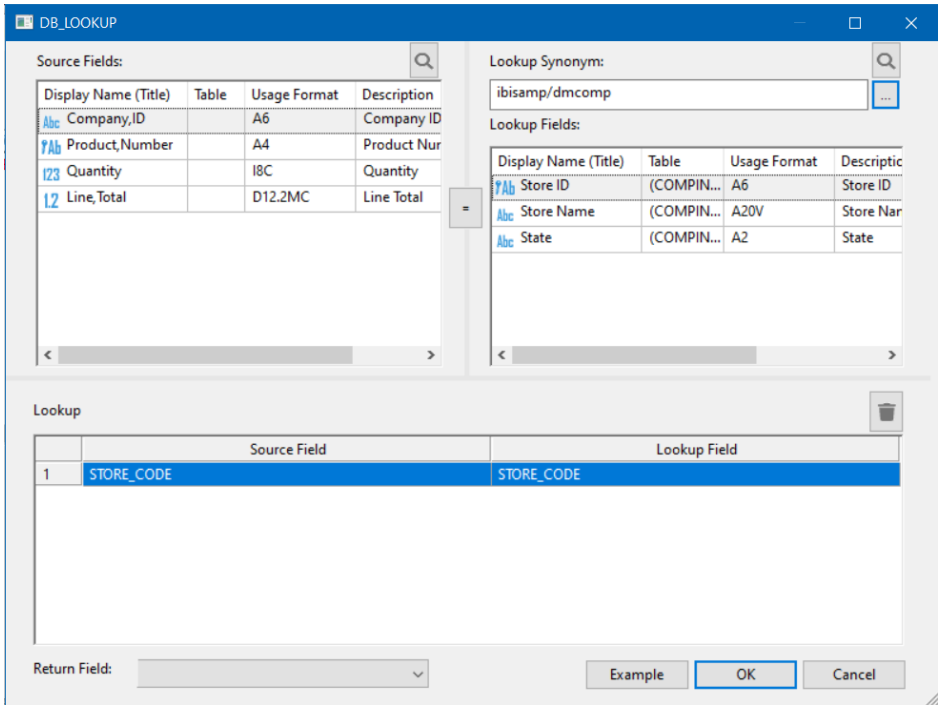
    The prototype for DB_LOOKUP appears in the Expression window and the Lookup dialog box opens.

15. Select a synonym to use for the lookup by clicking the ellipsis  button after the Lookup Synonym field.

    The Select Synonym dialog box opens.

16. Select the *DMCOMP* synonym and click *Select*.

17. When you have selected the columns, the Lookup window should appear, as shown in the following image.



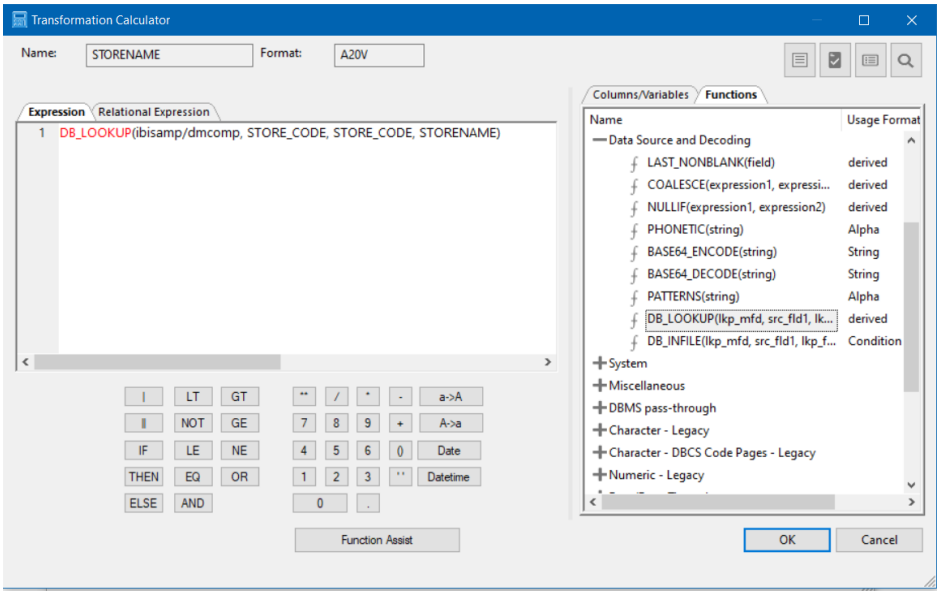
18. For *Return Field*, select *Store Name*.

19. Click *OK* to close the Lookup window.

20. Click *OK*.

The expression is completed in the Transformation Calculator window.

The Transformation Calculator should now appear, as shown in the following image.



21. Click *OK*.

    The STORENAME source transformation is added to the Transformations dialog box.

22. Repeat the process to add another lookup, using the parameters listed and described in the following table:

Parameter	Column Name
Lookup Synonym	DMINV
Source Column	PROD_NUM
Lookup Column	PROD_NUM
Return Value	PRODNAME

23. Click *OK* to close the Source Transformations dialog box.
24. Save the data flow as *DMXLKUPS*.
25. On the Flow tab, in the Run group, click *Run*, and then select *Run* from the drop-down menu.

When the data flow run is complete, verify that the data was loaded properly by right-clicking the *DMSTORES* object, selecting *Expand*, and clicking the *Sample Data* tab.

Note:

- The first four rows have a blank store name because the DMORD table contains store codes that do not exist in DMCOMP.
- Your output may vary depending on your database.

Record Logging

As you extract records from a data source and load them into a data target, a variety of record logging options are available. For example, you can log records that are rejected as duplicates, or records that are rejected due to format errors or when a field is truncated. The rejected records are written to a log file which can be used as input for another process. As this technique demonstrates, you can also write reports against these log files.

This technique is divided into two parts. In the first part, you create the log file. In the second, you view the records in the log file.

In the following example, we retrieve records from DMSALE, which contains sales representative information. The data source contains duplicate records, and these are rejected and written to a log file.

The location for the rejected records file is specified in the Record Logging tab of the target object, and we use the default, which will be dmxduplg_dupl. This allows you to easily display the records in the log file, which you will do in the second part of the technique.

Refer to the sample data flow DMDUPLG for the complete example.

Example: Creating a Log File and Viewing Records in the Log File

In the DMC:

1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the workspace, with the SQL object displayed.
2. Drag the data source object *DMSALE* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Right-click the SQL object and click *Column Selection*.

   The Column Selection window opens.
4. Select *SALESREP* and click the arrow to add it to the Selected Columns list. Then add *PLANT* to the Selected Columns list.
5. Click the *Distinct* check box, and click *OK*.

   This will create one row for each unique combination of SALESREP and PLANT.

6. Save the data flow as *DMXDUPLG*.

   Note: When you save, you will get a message warning you that the flow is incomplete, since you have not defined a target. Save the flow anyway.

7. Right-click the workspace to the right of the SQL object, select *Add Target*, and then click *New*.
8. In the Properties window, select the adapter you used for your sample procedures, the application directory you are using for your samples, and enter *DMSREP* as the synonym.
9. Under Target Load Options, select *Reject the record* from the If the record exists drop-down menu.
10. On the Flow tab, in the Tools group, click *Properties*. The Properties page opens.
11. Under Record Logging, select the *Duplicate Rejections* check box.

    Note that the rejected duplicates will be written to dmxduplg_dupl.ftm in the application directory you select for the target.

12. Click the *X* to close the Properties page.
13. Right-click the *DMSREP* target object and click *Target Transformations*.

    The Transformations window opens.

14. Check the *Key column* for *SALESREP* in the Target Columns list to designate a key column in the data target. Click *OK*.
15. Save the data flow.
16. On the Flow tab, in the Run group, click *Run* and select *Submit* from the drop-down menu.
17. Right-click *DMXDUPLG* in the navigation pane, select *Logs*, and then click *Last Log*.

The log of the data flow shows that 148 rows were processed, 126 rows were accepted and inserted into the data target, and 22 rows were rejected because of duplicates. The following image shows the log.

	Datetime	Message Code	Log Messages
1	2020/06/25 14:51:41	(ICM18974)	------ Start of Log Record for dmgeneral/dmduplg ------
2	2020/06/25 14:51:41	(ICM18122)	Request - dmduplg (Owner: dm) submitted.
3	2020/06/25 14:51:41	(ICM18741)	dmgeneral/dmsrep type Focus/FDS New target
4	2020/06/25 14:51:41	(ICM18429)	Issuing PREPARE
5	2020/06/25 14:51:41	(ICM18054)	Issuing CREATE and DROP TABLE for dmsrep
6	2020/06/25 14:51:41	(ICM18743)	Starting Load
7	2020/06/25 14:51:41	(ICM18682)	Check set to : 0
8	2020/06/25 14:51:42		0 WARNING..TRANSACTIONS ARE NOT IN SAME SORT ORDER AS FOCUS FILE
9	2020/06/25 14:51:42		PROCESSING EFFICIENCY MAY BE DEGRADED
10	2020/06/25 14:51:42	(ICM18041)	-- stats for source file
11	2020/06/25 14:51:42	(ICM18516)	148 : Row(s) processed by job
12	2020/06/25 14:51:42	(ICM18519)	0 : Row(s) rejected due to format error
13	2020/06/25 14:51:42	(ICM18372)	-- stats for target file: dmgeneral/dmsrep
14	2020/06/25 14:51:42	(ICM18516)	148 : Row(s) processed by job
15	2020/06/25 14:51:42	(ICM18514)	126 : Row(s) accepted by target table
16	2020/06/25 14:51:42	(ICM18515)	126 : Row(s) inserted into target table
17	2020/06/25 14:51:42	(ICM18517)	0 : Row(s) updated in target table
18	2020/06/25 14:51:42	(ICM18518)	0 : Row(s) deleted from target table
19	2020/06/25 14:51:42	(ICM18520)	0 : Row(s) rejected due to validation errors
20	2020/06/25 14:51:42	(ICM18521)	0 : Row(s) rejected due to no match
21	2020/06/25 14:51:42	(ICM18522)	22 : Row(s) rejected because duplicate exist
22	2020/06/25 14:51:42	(ICM18808)	0 : Row(s) rejected due to DBMS error
23	2020/06/25 14:51:42	(ICM18744)	Ending Load
24	2020/06/25 14:51:42	(ICM18040)	Return Code = 0
25	2020/06/25 14:51:42	(ICM18076)	Request: dmduplg - finished processing
26	2020/06/25 14:51:42	(ICM18007)	CPU Time : 16
27	2020/06/25 14:51:42	(ICM18031)	Finished

Example: Viewing Records in the Log File

After the DMXDUPS data flow created in the last example has run:

1. Right-click the application directory that you specified in the Record Logging tab during the previous step, and click *Refresh*.
2. Right-click the *dmxduplg_dupl* object from the application directory in the navigation pane and click *Sample Data*.

The records that are displayed in the following image are the 22 duplicate records that were rejected and placed in the dmxduplg_dupl log file.

	Sales Rep		Plant Locatio
1	Jacky	Li	BOS
2	Andreas	Graff	BOS
3	Yafa	Yardeni	ORL
4	Deigo	Castano	DAL
5	Deiter	Wolf	ORL
6	Niklas	Schwieler	STL
7	Goran	Hagman	STL
8	Ian	Bramley	DAL
9	Alex	McKay	STL
10	Guenter	Luebke	BOS
11	Bjorn	Danielson	DAL
12	Daniel	Hagman	BOS
13	Karl-Heinz	Zsenserling	BOS
14	Wilson	Fukushima	BOS
15	Per	Persson	BOS
16	Gil	Giati	DAL
17	Heinz	Gehri	STL
18	Patrick	Infante	LA
19	Eberhard	Steen	ORL
20	Anthony	Au	ORL
21	Christopher	Fox	ORL
22	Thomas	Schwieler	STL

Note: Your order may differ.

Using Variables in Data Flows

There are times when you want to move data and vary something in the data flow each time that it runs. You might want to only pull records for a particular range of dates, for a particular department, or any criteria that changes from run to run. Data Migrator lets you use a variable in the data flow to accomplish this.

The most common use of variables is in a filter. When the data flow is executed, the variable must be set to an appropriate value.

A value could also be set by executing a pre-extract stored procedure. For example, if the value of the variable is read from a fixed format file or other external source.

The following data flow extracts data from the DMORD table. The filter contains a variable for date, &ODATE, so that each time the flow is run &ODATE can be passed a different value, and records with different dates can be extracted from the data source and loaded into the data target.

The process flow calls the data flow passing a value for the variable as a parameter.

Refer to the sample data flow DMVAR and the sample process flow DMVARPF for the complete example.

Example: Using Variables in Data Flows

In the DMC:

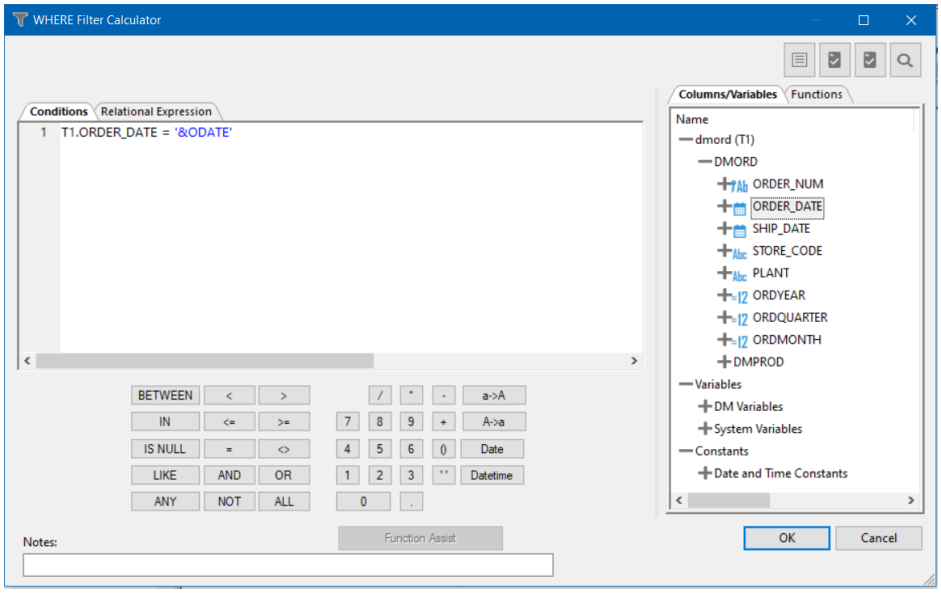
1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The Data Flow tab opens in the workspace, with the SQL object displayed.
2. Drag the data source object *DMORD* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Right-click the SQL object and click *Column Selection*.

   The Column Selection window opens.
4. Select *Plant,Location (PLANT)*, *Product,Number (PROD_NUM)*, and *QUANTITY* in the Available Columns list and click the arrow to move them into the Selected Columns list.
5. Select *T1.PLANT* and *T1.PRODNUM* in the Selected Columns list, and select *Group By* from the Aggregate drop-down menu.
6. Select *T1.QUANTITY* in the Selected Columns list, and select *Sum* from the Aggregate drop-down menu.
7. Click *OK* to close the Column Selection window.
8. Right-click the SQL object and click *WHERE Filter*.

   The Filter Calculator opens.
9. Double-click *Date of,Order (ORDER_DATE)* under Columns/Variables.
10. Click the *Conditions* tab.
11. Complete the condition by typing the following after T1.ORDER_DATE in the Conditions field:

```
= '&ODATE'
```

The Filter Calculator should appear, as shown in the following image.



12. Click *OK* to close the Filter Calculator.
13. Right-click the workspace to the right of the SQL object, select *Add Target*, and then click *New*.
14. Right-click Target object and click *Properties*. Select the adapter you used for your sample procedures. Then enter the application directory you are using for your samples dmplant as the name, for example, baseapp/dmplant. Then click the small *x* to close the properties panel.
15. Right-click the *DMPLANT* target object and click *Target Transformations*.

    The Transformations window opens.
16. Check that the *Key column* for *PROD_NUM* and *PLANT* is automatically selected in the Target Columns list to designate them as key columns in the data target. Click *OK*.
17. Save the data flow as *DMXVAR*.

Example: Prompting for a Variable Value

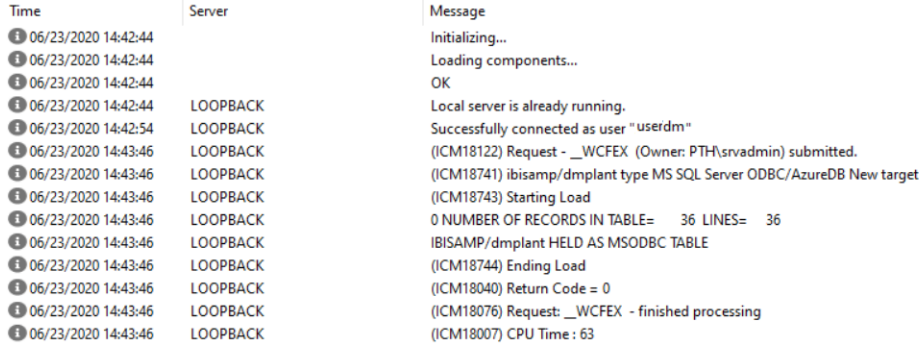
In order to test the data flow you need to supply a value for the variable.

1. On the Flow tab, in the Run group, click *Run*, and then select *Submit*.

   The prompting window opens. Enter a value to test, for example, 2006/02/14.

2. Click *Run*.

   The data flow is tested using the supplied value. The output from the flow appears in the console log window, as shown in the following image.



Time	Server	Message
06/23/2020 14:42:44		Initializing...
06/23/2020 14:42:44		Loading components...
06/23/2020 14:42:44		OK
06/23/2020 14:42:44	LOOPBACK	Local server is already running.
06/23/2020 14:42:54	LOOPBACK	Successfully connected as user "userdm"
06/23/2020 14:43:46	LOOPBACK	(ICM18122) Request - __WCFEX (Owner: PTH\srvadmin) submitted.
06/23/2020 14:43:46	LOOPBACK	(ICM18741) ibisamp/dmplant type MS SQL Server ODBC/AzureDB New target
06/23/2020 14:43:46	LOOPBACK	(ICM18743) Starting Load
06/23/2020 14:43:46	LOOPBACK	0 NUMBER OF RECORDS IN TABLE= 36 LINES= 36
06/23/2020 14:43:46	LOOPBACK	IBISAMP/dmplant HELD AS MSODBC TABLE
06/23/2020 14:43:46	LOOPBACK	(ICM18744) Ending Load
06/23/2020 14:43:46	LOOPBACK	(ICM18040) Return Code = 0
06/23/2020 14:43:46	LOOPBACK	(ICM18076) Request: __WCFEX - finished processing
06/23/2020 14:43:46	LOOPBACK	(ICM18007) CPU Time : 63

Example: Creating a Process Flow to Pass Parameter to a Data Flow

To create the process flow that runs both the stored procedure and the data flow:

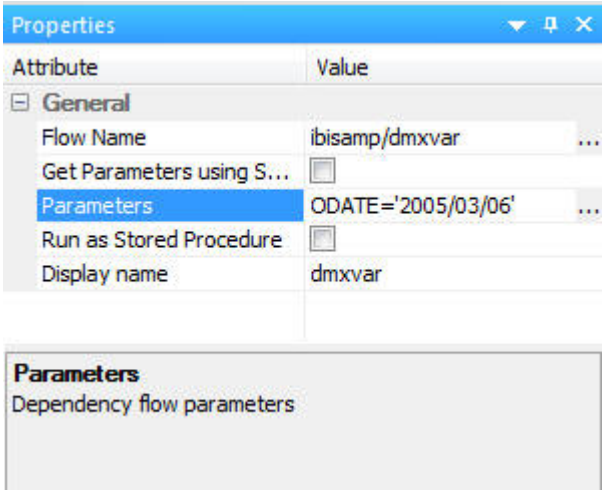
1. Right-click an application directory in the navigation pane, select *New*, and then click *Flow*.
2. Click the *Process Flow* tab.
3. Drag the data flow, *DMXVAR*, from the navigation pane into the process flow workspace, to the right of the Start object.
4. Right-click the *Start* object and drag to the *DMXVAR* object to connect them.
5. Right-click the *DMXVAR* object and click *Properties*. Click the ellipsis  button for Parameters.

   The Parameters Editor opens.
6. Click the *Add Parameter* button.

   The Drill Down Parameter editor dialog box opens.
7. Select *ODATE* from the Parameter Name drop-down menu.
8. Enter a date value, such as 2005/03/06 in the box provided.
9. Click *OK*.

   The parameter appears in the Parameter Editor.
10. Click *OK*.

The parameter appears in the property pane, as shown in the following image.



11.Save the flow as *DMVARPFX*. You can submit the flow now.

Creating a Data Flow Using a Target Based on a Predefined XML Schema

When creating an XML file, the format of the XML file is usually predefined by an xml schema definition or .xsd file. Data Migrator lets you create a synonym from a schema definition and then map to that target in the same manner as a relational database table. An existing JSON document can also be used as a prototype for a synonym, which can then be used as a target.

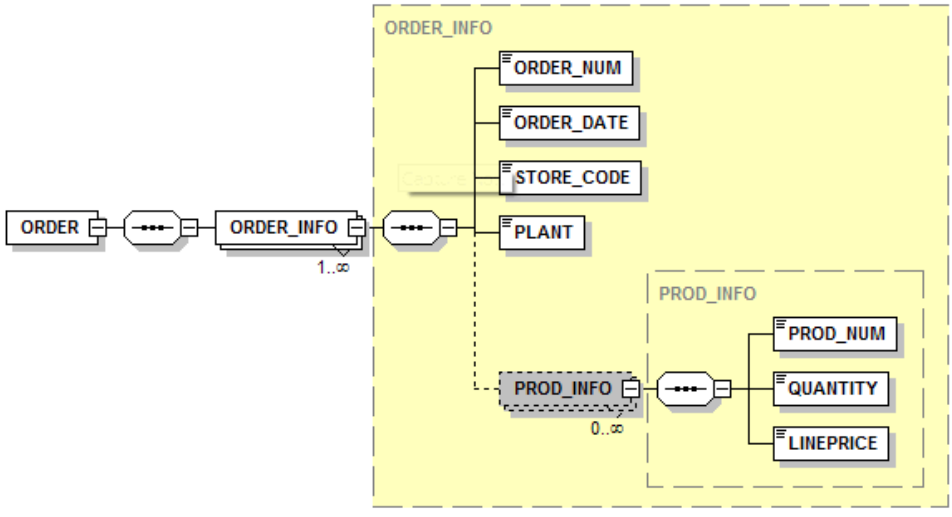
Single-segment and multiple-segment schemas are supported.

This technique is divided into two parts. In the first part, you create a synonym for the XML file. In the second, you use that synonym as the target of a flow.

In the example, the data source is DMORD. The target is described by the schema dmxord.xsd.

Refer to the sample data flow DMORDXML, for the complete example.

The following image shows the structure of the schema (dmxord.xsd), which will be used to create a synonym to demonstrate this technique.



Note: To write to an XML target, the XML adapter must be configured on your server.

Example: Creating a Synonym for an XML File

An XSD (XML Schema Definition) file contains a description of an XML file. This example shows how to create a synonym for the dmxord.xsd, which is installed with your server.

Note: For more information about creating synonyms for XML, please see *Using the Adapter for XML* in the *TIBCO WebFOCUS® Adapter Administration* manual.

1. In the DMC, right-click the *baseapp* application directory in the navigation pane, point to *New*, and then click *Synonym (Create or Update)*.

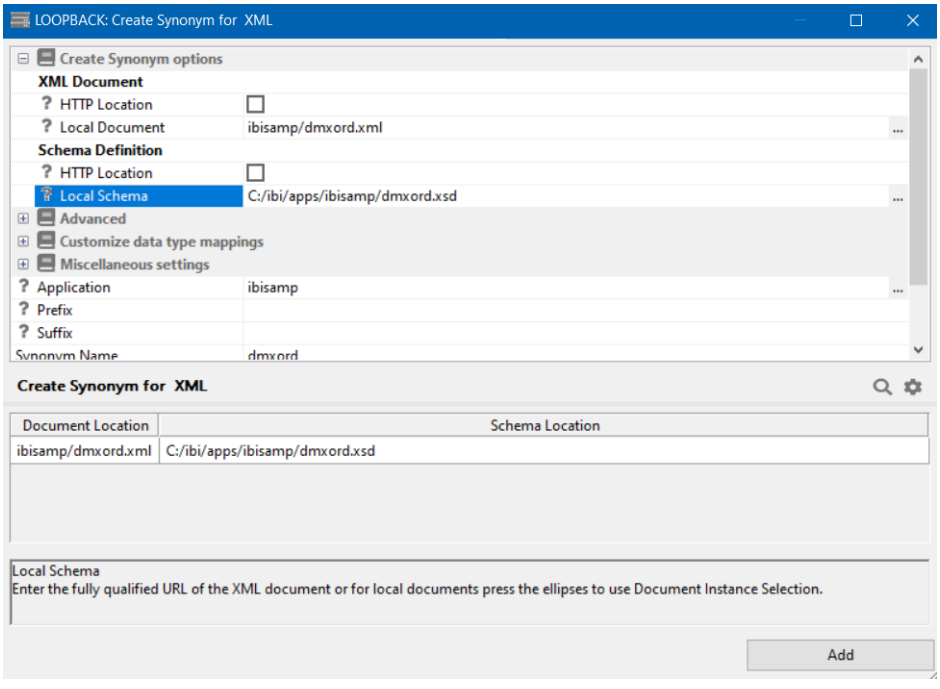
   The Select adapter to configure or Select connection to create synonym dialog box opens.

2. Right-click *XML* and click *Configure*.
3. Right-click XML on the Adapters list, and click *Create Metadata Objects*.

   The Create Synonym for XML dialog box opens.

4. Enter ibisamp/dmxord.xml as the location where the XML file is created.

5. Click the ellipsis to open Local Schema. Expand the *Applications* folder and *ibisamp*. Select *dmxord.xsd* and click *OK*. The name ibisamp/dmxord.xsd appears as the location of the XML Schema Definition that describes the xml target, as shown in the following image.



6. Click *Add*.
7. Click *Close* in the confirmation window.

Example: Creating an XML File

For a complete example, see the ibisamp/dmordxml flow.

In the DMC:

1. Right-click an application directory in the navigation pane, select *New*, and then click *Direct Load Flow*. The Direct Load tab opens.

   Right-click the flow canvas and click *Flow Properties*. In the Properties panel, deselect *Optimize Load*.
2. Drag the *DMORD* object from the ibisamp application directory in the navigation pane into the workspace, to the left of the line.
3. Drag the *DMXORD* synonym from the baseapp application directory in the navigation pane into the workspace, to the right of the line.

4. Double-click the *DMXORD* object.

   The Transformations window opens.

5. Click the *Automap* button.

6. Select the first row in Expressions panel, *ORDER_NUM*, and click the *Adjust Expressions* button. In the XML target, the ORDER_NUM column is described as Decimal. This expression converts the character-valued column to a number.

7. Click *OK*.

8. On the Flow tab, in the Run group, click *Run* and select *Submit*. A Save Changes dialog box opens. Click *Yes*.

9. When the Save Procedure as dialog box opens, enter *dmxord2xml* in the File Name field and click *OK*. Wait for the request to complete.

10. On Windows, you can view c:\ibi\apps\baseapp\dmxord.xml in Internet Explorer or in an XML editor. In the example shown in the following image, note that the hierarchical structure of the source has been preserved. There is one ORDER_INFO element which contains multiple PROD_INFO elements.



Loading a Target Table Using the File Listener

You can use Data Migrator to build a flow that automatically processes source files as they appear in a designated directory. This can be useful if, for example, source files are transferred by FTP from various locations throughout the day and need to be processed immediately.

Before you create and run the data flows discussed in this section, you must create the sample data and synonyms. For more information, see *How to Create Sample Procedures and Data for Iteration* on page 63. This will create source and target synonyms and four source data files used in the example.

To implement and run a real time flow:

1. Create an application directory to receive files.
2. Create a synonym with File Collection Parameters.
3. Create a Data Flow that uses the File Listener.
4. Run the flow and drop the files.

This fixed format file synonym corresponds to the layout of the source files that will appear in the designated directory. The fixed format file source is loaded to the target of your choice in real time.

Refer to the sample flow DMFLORD in the ibisamp directory for the completed example.

Example: Create Listen and Archive Directories

For Data Migrator to automatically process files when they arrive in a directory, you need to designate a directory for that purpose. It could be any directory that your server can access. For this example, we will create an application directory called listen.

In addition, if you want to keep a copy of the files after they are processed you need to designate a directory for that purpose too. For this example, we will create an application directory called archive.

1. Right-click *Application Directories*, select *New*, and then click *Application Directory*.

   The Create New Application dialog box opens.

2. For Application name, type *listen*.

   De-select the Add directory to APPPATH check box.

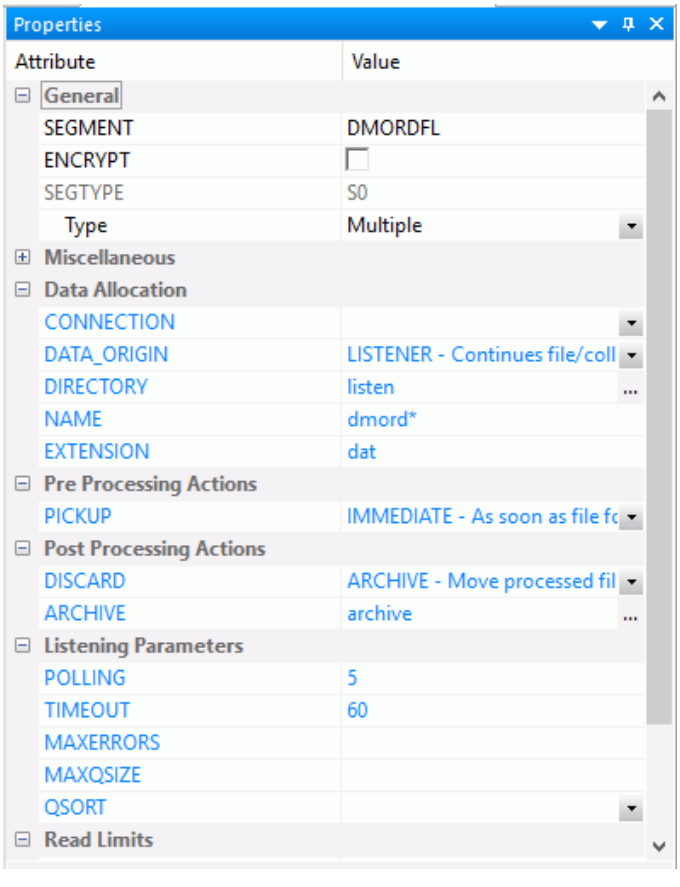
   Click *OK*.

3. Repeat steps 1 and 2 to create a directory called archive.

Example: Create a Synonym With File Collection Parameters

Next you need a synonym with File Collection Parameters to control how the file listener processes files. We will start with a synonym that describes just the record layout of the source file, and add the parameters.

1. In the navigation pane, expand the *Application Directories* folder, and then expand the *ibisamp* folder.
2. In the ibisamp folder, double-click *dmordflx* to open the synonym. Note that the column named SOURCEFILE has a special ALIAS of INSTANCE. This field does not exist in the source file itself, but it will contain the name of the source file when it is read.
3. Expand the *Data Allocation and other related* groups.
4. Fill in the synonym parameters, as shown in the following image.



This creates a synonym that listens on the listen directory for files with names that start with dmord and an extension of data polling every 5 seconds until 60 seconds have passed without a new file. As files are processed they are archived to a directory archive. Files are processed as they arrive until 5 files or 50 records are processed.

5. Click *Save As* in the Quick Access Toolbar to save the synonym in the application directory of your choice.

Example: Create a Flow with a File Listener Source

A flow with a File Listener source can be either a Data Flow or a Direct Load Flow. For this example, use a Direct Load Flow.

In the DMC:

1. Right-click an application directory, point to *New*, and then click *Direct Load Flow*. The Direct Load Flow opens.
2. From the browser window, drag the source object *DMORDFL* that you created from its application directory to the left of the center line in the workspace.
3. Drag the target object *DMORDF* from the ibisamp application directory to the right of the target line in the workspace.
4. Right-click the target object *DMORDF* and click *Properties*.

   The Properties dialog box opens.
5. For Prior to Load Option, select *Truncate table*, if available. If not, select *Delete all rows from table* or *Drop table*. Click *X* to close the Properties page.
6. Right-click the target object and click *Target Transformations*.

   The Transformations window opens.

   Click the *Automap* button and click *OK*.
7. Save the flow as *DMXFLORD*.

Example: Create a Synonym for a File Listener with a Trigger

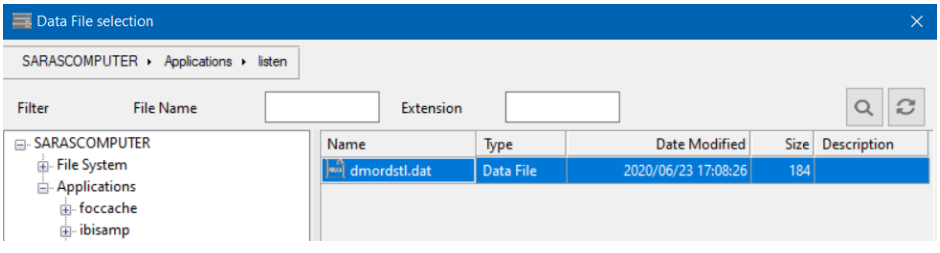
You can use Data Migrator to build a flow that automatically processes source files as soon as a specific Trigger file is detected in an application directory. A trigger is typically used when processing needs to be delayed. For example, when using FTP to copy files, if you wait until this secondary file appears while a large file is being copied into a directory, the trigger file indicates when processing can begin.

The following example shows how to create such synonym in the baseapp application using an Adapter for Delimited Files. You can also create such a synonym using the Adapter for a Fixed Format File or in any other application directory.

1. Create Listen and Archive directories as described in the *Create Listen and Archive Directories* example.
2. In the DMC, right-click the *baseapp* application directory, point to *New*, and then click *Synonym (Create or Update)*.

   The Get Data dialog box opens.
3. Right-click the Delimited File node and click *Show local files*.

   The Select Data File dialog box opens,
4. Select the data file from the Listen directory, as shown in the following image.



   Data Migrator creates the Master File for the File Listener synonym for the synonym later, based on this file.
5. The Create Synonym dialog box opens.

   Enter the following values in the Advanced section:

   a. Select *Collection of files*.

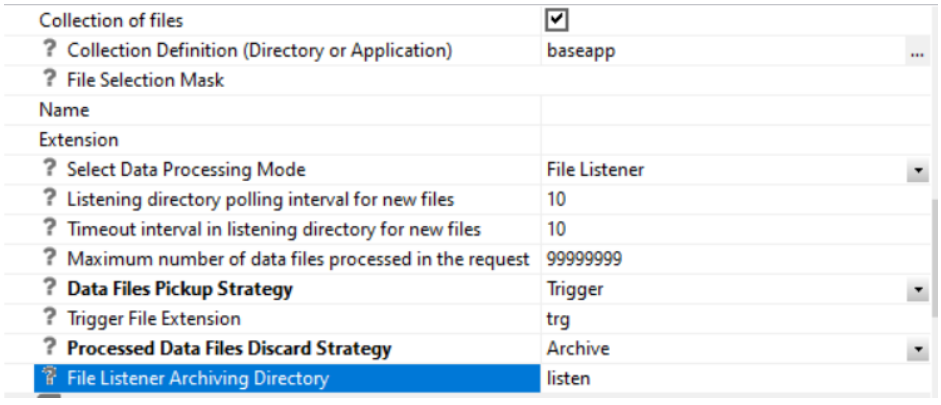
   b. Select *Trigger* from the Data File Pickup Strategy drop-down menu.

   c. Enter the extension in the Trigger File Extension field. For example, if the file name that you listen to is myfile.csv, the trigger file name is myfile.csv.trg. The value for the Trigger File Extension will be *trg*.

   d. Select *Archive* from the Processed Data Files Discard Strategy drop-down menu.

   e. Enter the name of your Archive directory in the File Listener Archiving Directory field.

The options should appear as shown in the following image.



6. Enter all attribute values that are needed for the Delimited File.
7. Click *Add*.

Example: Running a Flow With a File Listener Source

In this example, we will start the flow and then copy source files to the listen directory.

In the DMC:

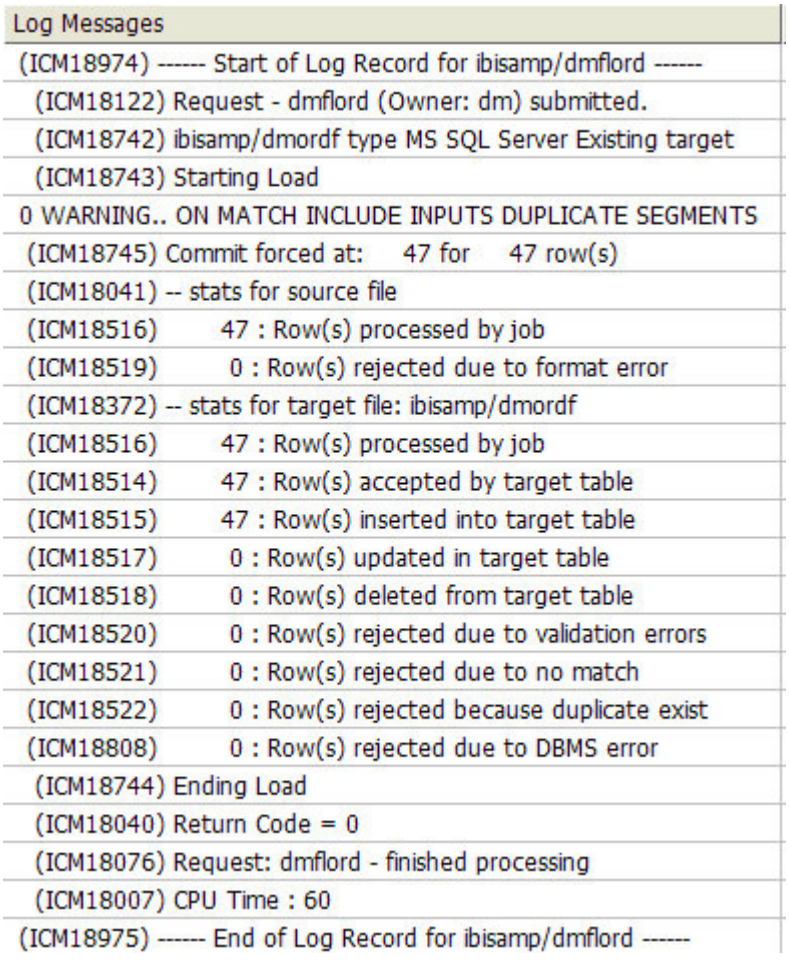
1. Right-click *DMXFLORD* and click *Submit*.
2. Copy the following files, one at a time, from the ibisamp application directory to the listen application directory:

   dmordbos.dat, dmorddal.dat, dmordorl.dat, dmorstl.da

   Tip: For a Windows based server, you can use the Edit and Copy/Paste functions.

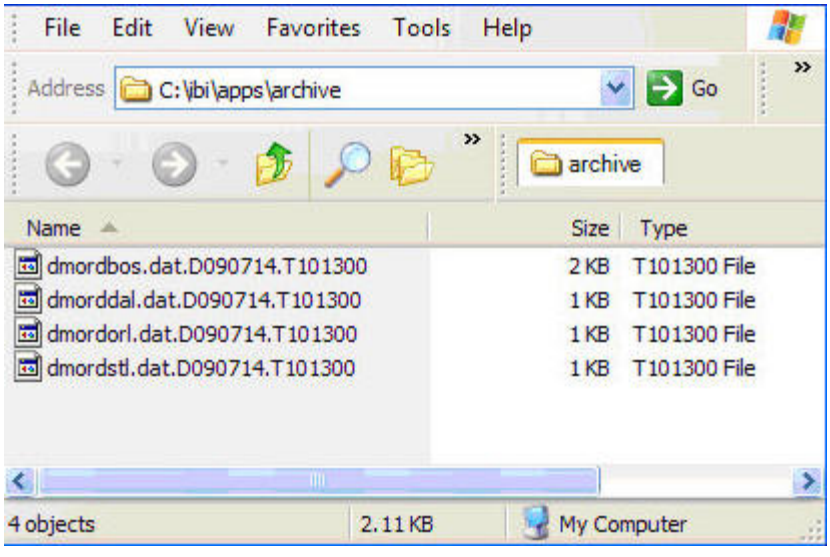
   Note: As the files are processed, you should see them disappear from the listen directory. This could take up to the timeout interval specified in the synonym.
3. Return to the DMC. The source synonym has a timeout of 60 seconds, so you will see the Processing dialogue until the time has elapsed.

4. When the flow completes, on the Flow tab, in the Reports group, click *View Last Log*. You should see a report, like the following image, that shows all total number of rows processed.



Log Messages
(ICM18974) ------ Start of Log Record for ibisamp/dmflord ------
(ICM18122) Request - dmflord (Owner: dm) submitted.
(ICM18742) ibisamp/dmordf type MS SQL Server Existing target
(ICM18743) Starting Load
0 WARNING.. ON MATCH INCLUDE INPUTS DUPLICATE SEGMENTS
(ICM18745) Commit forced at: 47 for 47 row(s)
(ICM18041) -- stats for source file
(ICM18516) 47 : Row(s) processed by job
(ICM18519) 0 : Row(s) rejected due to format error
(ICM18372) -- stats for target file: ibisamp/dmordf
(ICM18516) 47 : Row(s) processed by job
(ICM18514) 47 : Row(s) accepted by target table
(ICM18515) 47 : Row(s) inserted into target table
(ICM18517) 0 : Row(s) updated in target table
(ICM18518) 0 : Row(s) deleted from target table
(ICM18520) 0 : Row(s) rejected due to validation errors
(ICM18521) 0 : Row(s) rejected due to no match
(ICM18522) 0 : Row(s) rejected because duplicate exist
(ICM18808) 0 : Row(s) rejected due to DBMS error
(ICM18744) Ending Load
(ICM18040) Return Code = 0
(ICM18076) Request: dmflord - finished processing
(ICM18007) CPU Time : 60
(ICM18975) ------ End of Log Record for ibisamp/dmflord ------

With the Discard option set to Archive, after each file is processed, it is copied to the archive directory, as shown in the following image.



These files can be deleted, as they are no longer needed.

Sending a Target File to an FTP or SFTP Server

You can use Data Migrator to build a flow that automatically sends the target file to a directory on a FTP server. This can be useful if, for example, a target file has to be created on a remote FTP location. This feature is available for the synonyms of a Fixed Format File and a Delimited File. The synonym can be used in a Direct Load flow, as well as in a data flow.

The examples in this section use a synonym for a Fixed Format File in a Direct Load flow.

Before you create and run the data flow discussed in this section, you must configure the Adapter for a Fixed Format File. After the adapter is configured, you need to create the sample data and synonyms for Data Migrator - General. This will create the source and target synonyms that are used in the following examples.

To implement and run a flow that sends a target to a FTP server, you must:

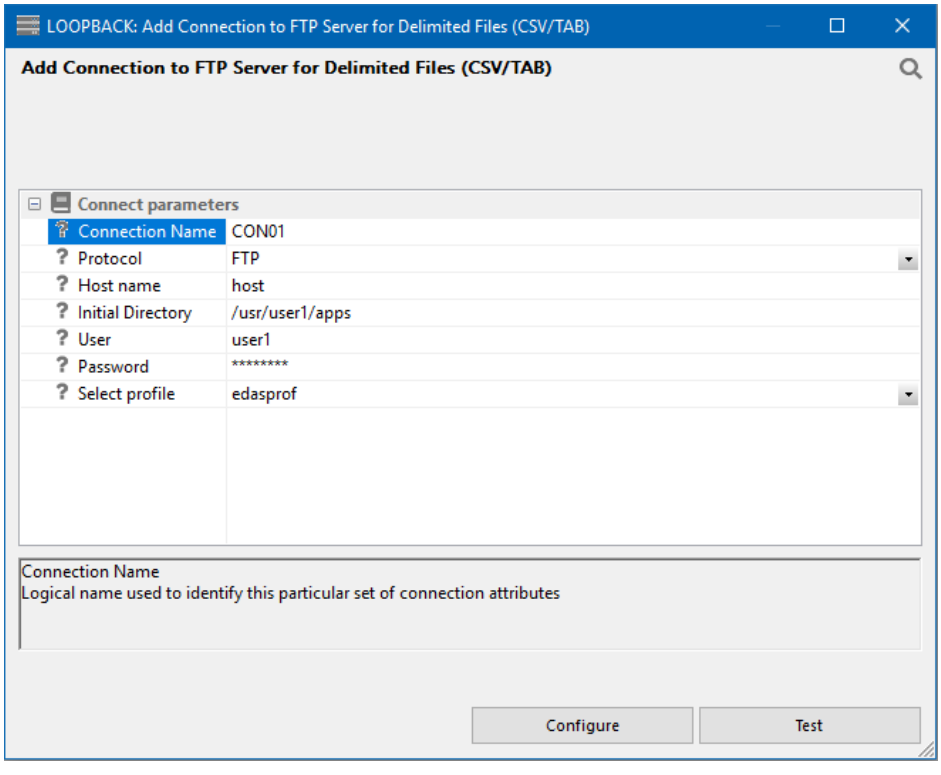
1. Create a connection to a FTP location.
2. Create a synonym with FTP attributes.
3. Create a Direct Load flow that uses this synonym as a target.

Example: Creating a Connection to a FTP or SFTP Location

Note: If you are using SFTP, you must first set up an SSH connection without a password. For more information, see *Using Secure Shell FTP (SFTP)* on page 78.

To configure a connection to a FTP server:

1. Expand the *Adapters* folder, followed by the folder.
2. Right-click the *Fixed Format File* or *Delimited File* folder and click *Add Connection*.
3. Enter the connection parameters, as shown in the following image.



Host Name

Name or IP address of the FTP or SFTP server.

Initial Directory

Directory on the FTP or SFTP server to connect to. This value is optional. If it is not specified, either of the following will be used:

- The default initial directory.

- The directory name entered on a synonym for a DATASET parameter value, entered with a file name, such as mydir/myfile.txt.

Note: On z/OS, to access files in zFS, enter a directory name. To access sequential datasets, enter two periods (..) or a prefix such as 'USERID.'.

User

User ID to connect to the FTP or SFTP server.

Password

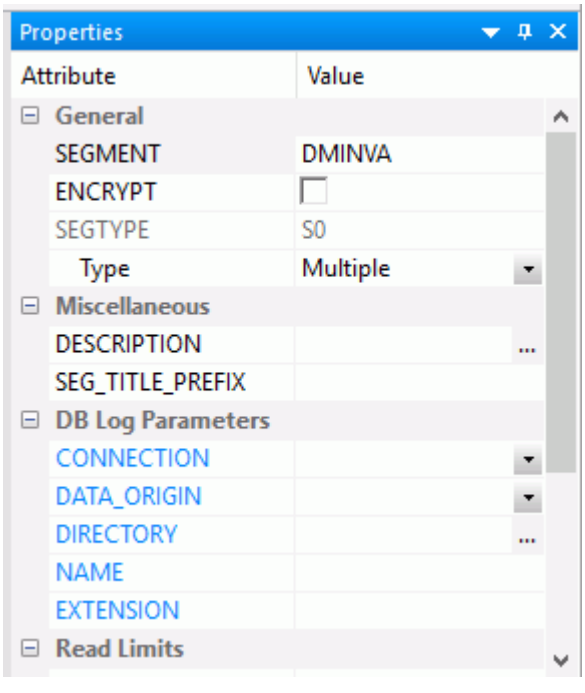
Password to connect to the FTP server.

Protocol

The file protocol that is being used, either FTP or SFTP (SSH FTP).

Example: Creating a Synonym with FTP Attributes

1. Open the synonym *dminva* in the Synonym Editor, as seen in the following image.



2. Enter the file name in the DATASET parameter.

3. In the DB Log Parameters section, enter the Connection value that was created for the Fixed Format File adapter.

   Leave the *DATA_ORIGIN* parameter blank.

4. Enter values for the following fields:

   - **DIRECTORY.** The location of the data files for polling.
   - **NAME.** The data file name.
   - **EXTENSION.** The data file extension.

5. Save the synonym with the new name *dminva_ftp*.
6. Close the synonym.

Example: Creating a Flow that Sends a File to an FTP Server

1. Right-click an application directory, select *New*, and then click *Direct Load Flow*.

   The Direct Load flow opens.

2. From the browser window, drag the *dminv* source object from the application directory to the left of the center line of the workspace.
3. Drag the *dminva_ftp* target object that was created from the ibisamp application directory to the right of the target line in the workspace, as shown in the following image.



4. Right-click the target object and click *Target Transformations*.

   The Transformations window opens.

5. Click the *Automap* button.
6. Save the flow as *dminvftp*.

Loading a Star Schema

A database in a data warehouse is often organized into a star schema, consisting of a central fact table with the data to be analyzed and multiple dimension tables that describes the data.

Each dimension table has a single surrogate key, an arbitrary unique identifier for the row. The fact table has multiple keys, each joining to a different dimension table.

Before you create and run the data flows discussed in this section, in addition to creating the source tables, you must also create the target tables. See *How to Create Sample Procedures and Data for Star Schema* on page 62.

This example has detailed instructions for loading one of the four dimension tables. Refer to the sample flows, DLOADCUST, DSLOADPROD, DSLOADSALE, and DSLOADTIME, for other examples of loading a dimension table. Refer to DSLOADFACT for an example of loading a fact table and DSFLOWP for the complete example of loading a star schema.

Example: Loading a Dimension Table

This example has instructions for loading the customer dimension table.

1. In the DMC, right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The data flow opens in the right hand pane, with the SQL object displayed.
2. Drag the data source object *DMCOMP* from the ibisamp application directory in the navigation pane into the workspace, located to the left of the SQL object.
3. Right-click the SQL object and click *Column Selection*.

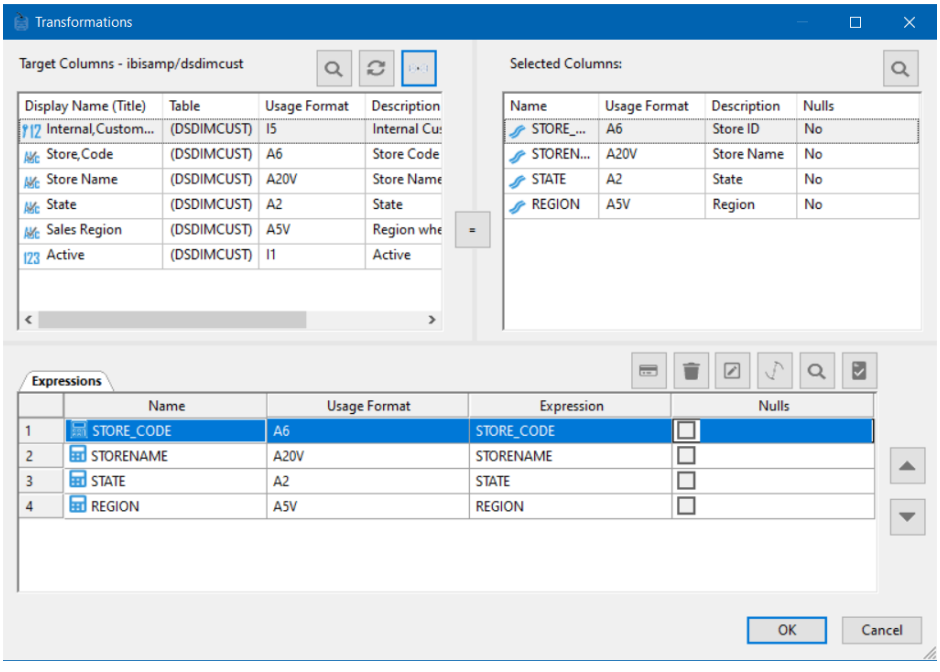
   The Column Selection window opens.
4. Right-click any column name and *Select All*. Then click the arrow to add them to the selected columns and click *OK*.
5. Drag the target object *DSDIMCUST* from the ibisamp application directory into the workspace to the right of the SQL object.
6. Right-click the target object *DSDIMCUST* and click *Properties*.

   The Properties page opens.

   From the *Load Type* drop-down menu, select *Slowly Changing Dimensions*. Click *X* to close the Properties pane.
7. Right-click the target object *DSDIMCUST* and click *Target Transformations*.

   The Transformations window opens.
8. Click the *Automap* button.

The identically named source and target column names are added to the Expressions window. The Transformations window should now look like the following image.



Note: The columns CKEY and ACTIVE are not mapped because these columns are handled automatically by the Data Migrator Slowly Changing Dimension processing. CKEY is the surrogate key, which automatically starts with a value of 1 and increases by one, for each row added. The Active flag is set to 1 for currently active rows.

Click *OK* to close the Transformations window.

9. Click *Save* from the Quick Access Toolbar. Enter *dsxloadcust* as the file name.

Example: Loading a Fact Table

A fact table load requires looking up the keys for each row in the corresponding dimension tables, and obtaining a surrogate key so that the fact table can be joined to each of the dimensions for subsequent reporting.

To create the fact table follow these steps.

1. In the DMC, right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The data flow opens in the right hand pane, with the SQL object displayed.

2. Drag the data source object *DMORD* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.
3. Drag the data source object *DMSALE* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.

   A JOIN object is automatically added connected to DMORD and DMSALE.

   Note: If the Join object does not automatically appear, delete the object. On the Home tab, in the Tools group, click *Options*. In the Options dialog box, click *Data Flow* and select the *Add Join Object if needed* check box. Click *OK*. Then add *dmsale* again.
4. Drag the data source object *DMINV* from the ibisamp application directory in the navigation pane into the workspace, to the left of the SQL object.

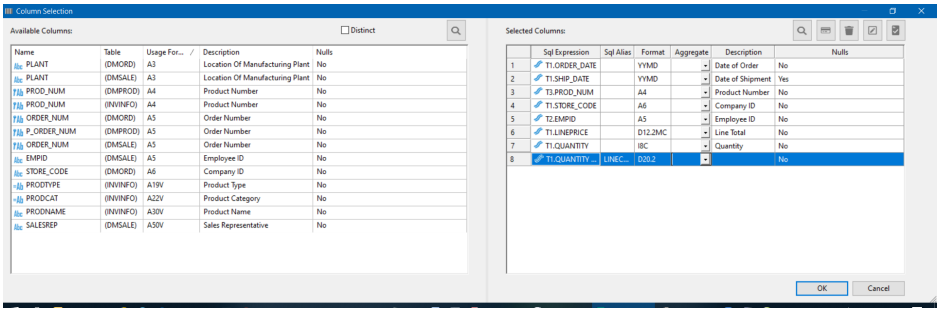
   A second JOIN object is automatically added connected to the first JOIN object and DMINV.
5. Right-click the SQL object and click *Column Selection*.

   The Column Selection window opens.
6. Select the following columns, and click the arrow to add them to the Selected Columns list:

   - Date of,Order (ORDER_DATE) from DMORD(T1)
   - Shipping,Date (SHIP_DATE) from DMORD(T1)
   - Product,Number (PROD_NUM) from DMINV(T3)
   - Company,ID (STORE_CODE) from DMORD(T1)
   - Employee,ID (EMPID) from DMSALE(T2)
   - Line,Total (LINEPRICE) from DMORD(T1)
   - QUANTITY from DMORD(T1)
7. Click the *Insert Columns* button.

   The SQL Calculator opens.
8. For Alias, type *LINCOGS* and for calculation, expand *DMPROD* from the Columns/Variables tab. Double-click *QUANTITY*, type*, then expand *DMINV* and *INVINFO*. Select *Our,Cost (COST)* so that the Expression window shows T1.QUANTITY * T3.COST.
9. Click *OK* to close the SQL calculator.

The Column Selection window should appear as in the following image.



10. Click *OK* to close the Column Selection window.
11. Drag the target object *DSFACT* from the ibisamp application directory to the workspace to the right of the SQL object.
12. Right-click *DSFACT* and click *Target Transformations*.

    The Transformations window opens.
13. Double-click *ODKEY*, adding it to the Expressions window. With the line selected, click the *Edit Transform* button.

    The Transformation Calculator window opens.
14. Click the *Functions* tab, expand the *Data Source and Decoding* folder, and double-click *DB_LOOKUP*.

    The Lookup function assist opens.
15. Click the ellipsis button after Lookup Synonym. Select the synonym *DSDIMTIME* from the ibisamp directory and click *Select*.
16. Select *ORDER_DATE* from the left hand side and *TDATE* on the right hand sideand click the = sign. Click *OK* to close the dialog box.
17. Select *TKEY* from the Return Field drop-down list.

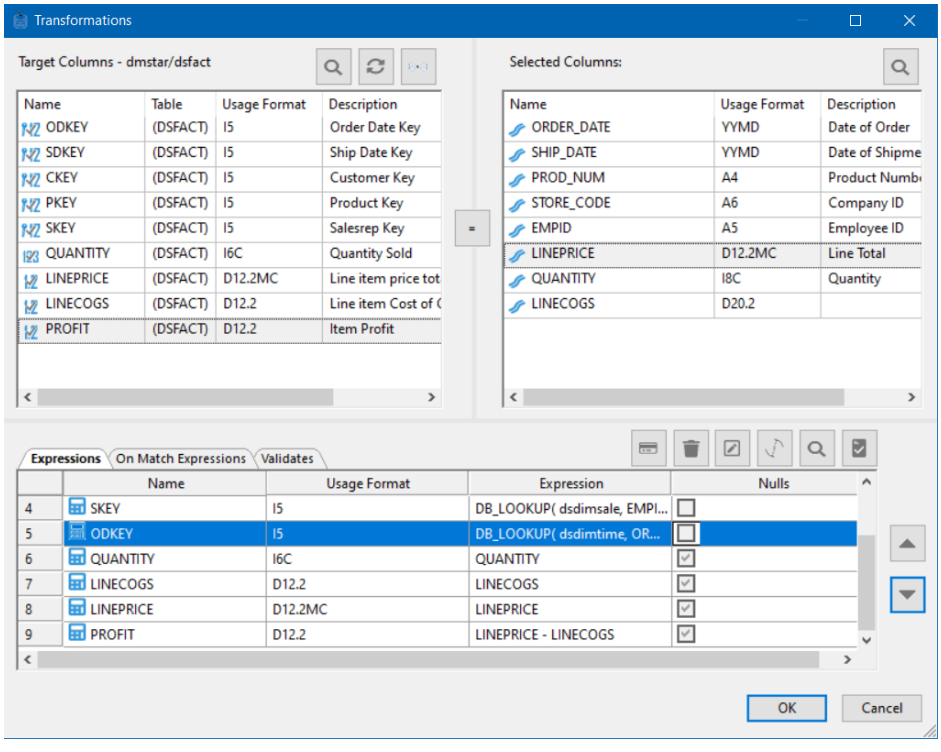
    Click *OK* to close the window.
18. Repeat steps 12-17 above for SDKEY, using SHIP_DATE as the source field, TDATE as the lookup field, and TKEY as the return field.
19. Similar transformations are required for each of the key columns.

    Use the following tables:

    CKEY - DSDIMCUST PKEY - DSDMINPROD SKEY - DSDIMSALE

For the rest of the key columns, the Source Field and Lookup Field are pre-selected, since the names are the same in the Fact and Dimension tables. In addition, transformations are required for the remaining columns.

When you have created the remaining transformations, the Transformations window should appear, as shown in the following image.



20. Click *OK* to close the window.
21. Click *Save as* from the Quick Access Toolbar. Enter *dsloadfact* as the file name.

Example: Loading a Star Schema Using a Parallel Group

In order to load the fact table in this example, all the dimension tables have to be loaded first, so that the surrogate keys are available.

1. In the DMC, right-click an application directory in the navigation pane, select *New*, and then click *Flow*. The data flow opens in the right pane.
2. Click the *Process Flow* tab. The view switches to the process flow with a Start icon.

3. On the Flow tab, in the Insert group, click *Parallel Group* and drag it into the work area to the right of the Start icon.

   A box appears on the screen. This is an empty parallel group.

4. Right-click the *Start* icon and drag a line towards the parallel group box.

   Release the line when the arrow is touching the box.

5. From the navigation pane, drag the *DSXLOADCUST* flow *inside* the parallel group box. If you did not create this data flow, drag the *DSLOADCUST* flow from the ibisamp directory.

6. From the ibisamp directory, drag the *DSLOADPROD*, *DSLOADSALE*, and *DSLOADTIME* flows into the parallel group.

7. The DSLOADTIME flow requires a parameter of the first date to load. Double-click the flow to open the properties for the flow. For Parameters, enter *STARTDATE=20040101*.

8. On the Flow tab, in the Insert group, click *Wait* and drag it into the work area to the right of the parallel group.

9. Right-click the parallel group, drag a line to the Wait icon, and release.

10. Drag the *DSXLOADFACT* flow to the right of the Wait icon. If you did not create this data flow, drag the *DSLOADCUST* flow from the ibisamp directory.

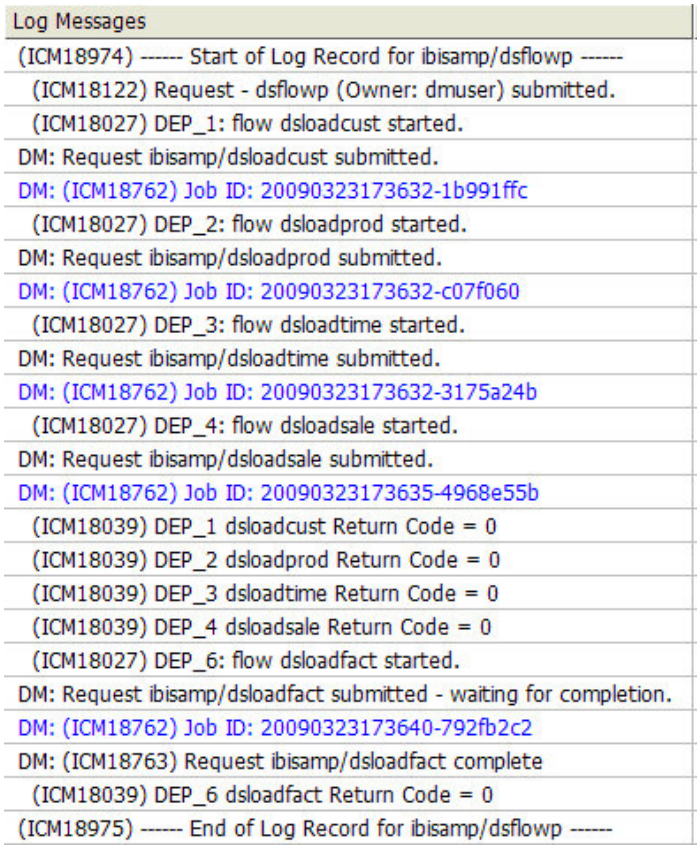
11. Right-click the *Wait* icon, drag the line to the DSLOADFACT flow, and release it.

12. Click *Save as* from the Quick Access Toolbar. Enter *dsxflowp* as the file name.

13. On the Flow tab, in the Run group, click *Run* and select *Submit* to run the flow.

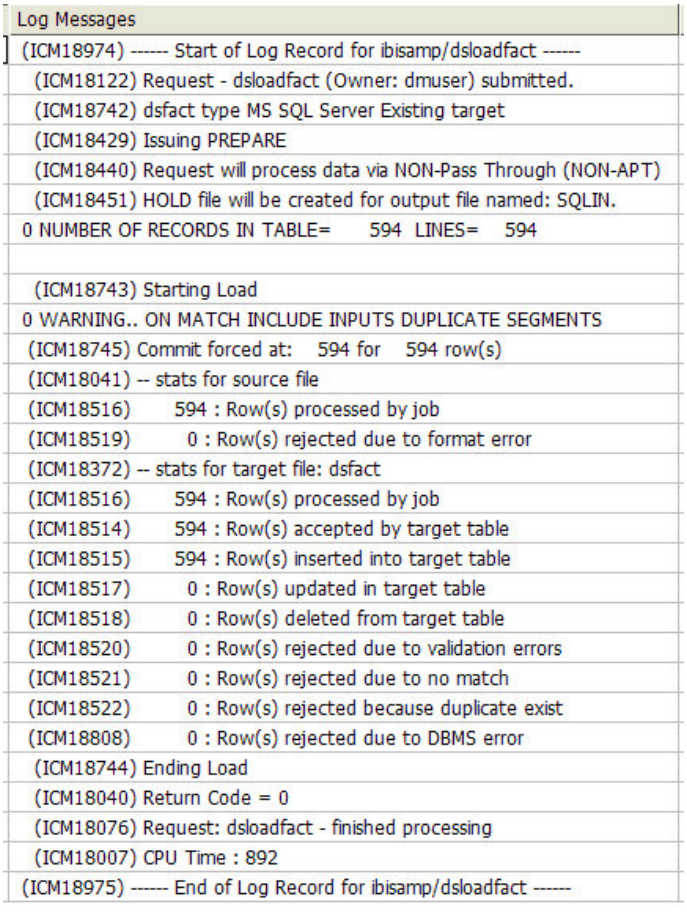
14. When the flow completes, on the Flow tab, in the Reports group, click *View Last Log*.

The process flow log opens, as shown in the following image.



The log shows the four dimension data flows run and the fact table load, which runs when the data flow runs are complete. The lines in blue indicate links to the detail logs for the individual flows.

15. Click the fact table log, which is the last blue line shown. The detail log opens, as shown in the following image.



Iterating Through Multiple Tables

It is often the case that an organization has multiple tables with the same structure, that is column names and data types, but with different names, in different databases or schemas or on different servers.

Now a single synonym can be used to describe a collection of tables with the same structure because the TABLENAME and CONNECTION parameters in the access file can be specified as a variable.

Before you create and run the data flows discussed in this section, in addition to the source data, you must also create the multiple source tables. See *How to Create Sample Procedures and Data for Iteration* on page 63.

This will create five identically structured tables named dmordbos, dmorddal, dmordla, dmordorl, and dmordstl.

Example: Create a Synonym With Parameterized Table Names

This example has instructions for creating a synonym that can be used to read or write multiple tables. You will create two variables, TTABLE and TCON, that will be used for the table name and connection name.

Refer to the sample synonym, dmortmpl, for the complete example that also has titles and descriptions for each of the columns.

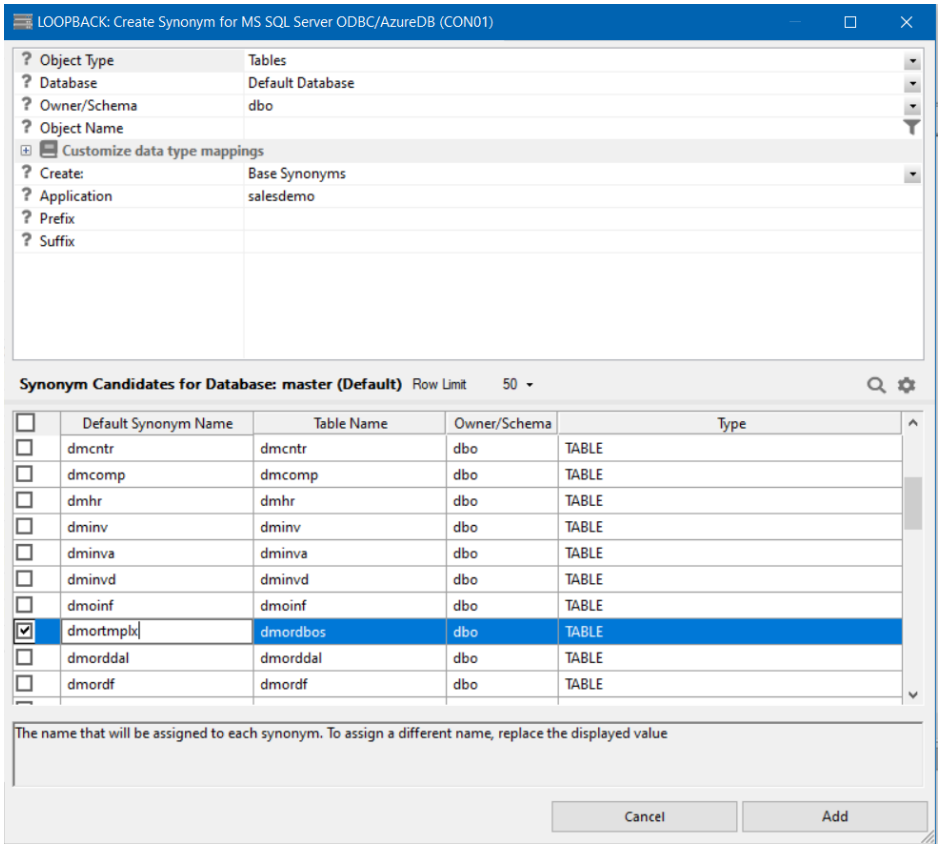
1. In the DMC, right-click an application directory in the navigation pane, select *New*, and then click *Synonym (Create or Update)*.

   The Get Data panel opens.

2. Right-click the connection for the adapter where you created the sample procedures, and click *Show DBMS Objects*.

   The Create Synonym panel opens.

3. Select the check box to the left of dmordbos, click in the field with the synonym name, and change it to *dmortmplx*, as shown in the following image.



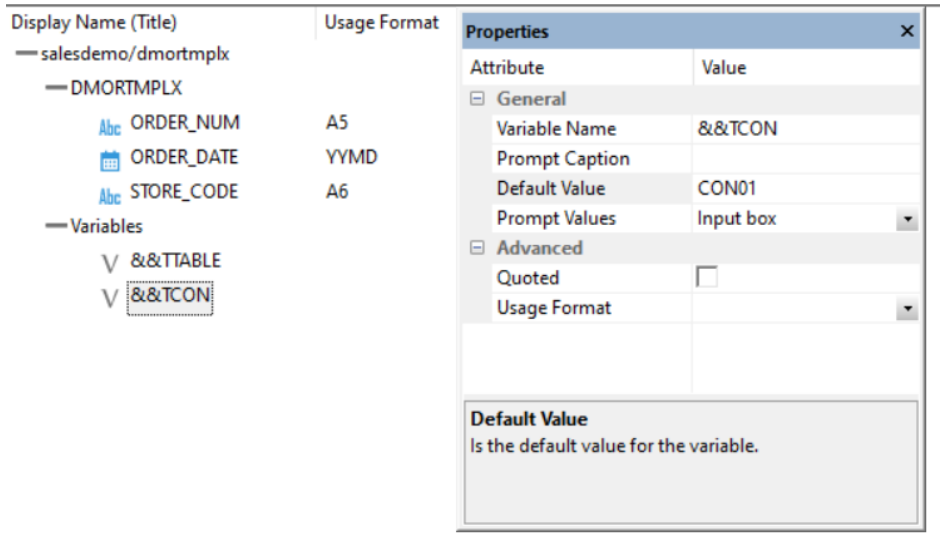
   This will create a synonym that you will use as a template to access any of the six tables.

4. Click *Add*.
5. In the Navigation pane, double-click dmortmplx.

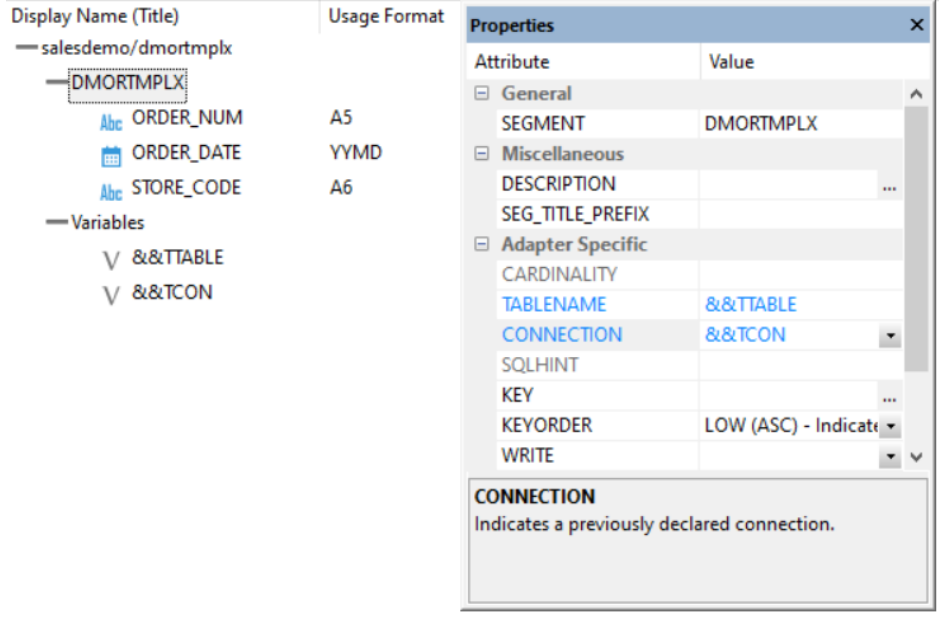
   The dmortmplx synonym opens.

6. If there is Business View window, click the small *x* to close it.
7. On the ribbon, click *Insert* and select *Variable*. This adds a variable called VARIABLE1.
8. Click the name and change it to *TTABLE*. This creates a global variable in the synonym.
9. For DEFAULT, enter the value *dmordbos*.
10. Right-click the *Variables* folder, select *Insert*, then *Variable*.
11. Click the name and change it to TCON. For the DEFAULT, enter the name of your first database connection, which by the server default would be CON01.

12.The names of the two variables you created are shown.



13.Click the segment name *DMORTMPLX*. For TABLENAME, enter the value *&&TTABLE* and for CONNECTION, enter the value *&&TCON*.



The synonym is now set up for use with multiple tables.

14.Click *Save* from the Quick Access Toolbar to save the synonym.

Example: View Parameter Table

This example has instructions for viewing the parameter table used in the other examples here. The parameter table used is a delimited file, but any data source that can be described by a synonym could be used. The table has three columns: a connection name, a table name, and a plant name.

The parameter table is automatically created when you run Create Sample Procedures and Data for the iterator examples. The connection name is the name of the first connection on your server for the selected database. In this example, they are all the same, but different connection names could also be used.

1. In the DMC, expand the *ibisamp* directory in the navigation pane.
2. Right-click the synonym *dmplnts* and click *Sample Data*.

   Tip: If you do not see synonyms when you expand the ibisamp directory, select *Synonyms* from the Filter group on the Home tab.

   The Sample Data Report opens and should look like the following image.

	LCON	LTABLE	LPLANT
1	CON01	dmordbos	BOS
2	CON01	dmorddal	DAL
3	CON01	dmordorl	ORL
4	CON01	dmordstl	STL
5	CON01	dmordla	LA

Example: Create a Data Flow Using a Parameterized Synonym

This example has instructions for creating a data flow that uses a synonym as a source that can be used to read multiple identically formatted tables.

Refer to the sample flow dmorsum, for the complete example.

1. In the DMC, right-click an application directory in the navigation pane, select *New*, and then click *Flow*. A new flow opens.
2. Drag a data source object into the data flow. If you created the synonym, dmortmplx, you can use it. Otherwise, from the ibisamp application directory, drag the synonym *dmortmpl*.

3. Right-click the SQL object and click *Column Selection*.

   The Column Selection window opens.

4. Select each column in the Available Columns list, and click the arrow to move them into the Selected Columns list. Click *OK*.

   Tip: To select all the columns, right-click one of them and click *Select All*.

5. Drag the target object *dmorsum* from the ibisamp directory into the workspace, to the right of the SQL object.

6. Right-click the *dmorsum* target object and click *Target Transformations*.

   The Transformations window opens.

7. Click the *Automap* button.

8. Under target columns, double-click *PLANT* to add it to the list of expressions below.

9. For the PLANT expression, enter *'&LPLANT'* with quotes as shown. Click *OK* to close the expression editor and *OK* to close the Transformations window.

10. Click the *Process Flow* tab.

11. Right-click the arrow between Start and Data flow and click *Delete*.

12. On the Flow tab, in the Insert group, click *Set Variables* and drag it onto the workflow between the Start and Data Flow icons. Then, right-click the *Start* icon and drag it to Set Variables and release it. Finally, right-click *Set Variables* and drag the arrow to Data Flow and release.

13. Right-click *Set Variables* and click *Properties*. In the Value area for List, click the ellipsis  button.

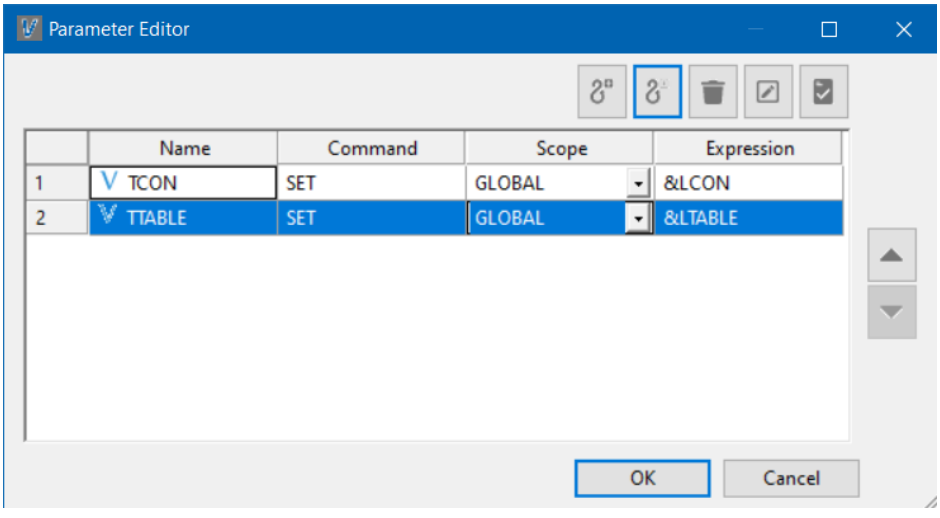
    The Global Parameter Editor window opens.

14. Click the *Add Parameter* button.

    The Global Variables calculator opens.

15. Type over the default name VAR1 with *TCON*. In the expression entry area, type *&LCON* and then click *OK*.

16. Repeat to set the global variable, TTABLE to value *&LTABLE*. When you are done, the Global Parameter Editor window should look like the following image.



Click *OK*.

17. Click *Save* from the Quick Access Toolbar to save the flow. Enter *dmorsumx* as the flow name.

Example: Create a Process Flow to Use Parameterized Data Flow

This example has instructions for creating a process flow that runs a data flow repeatedly using as source a collection of identically formatted tables.

Refer to the sample flow dmorsump for the complete example.

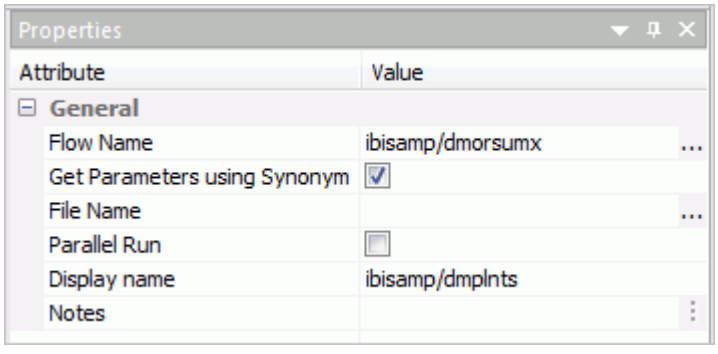
1. In the DMC, right-click an application directory in the navigation pane, select *New*, and then click *Flow*.

   A new flow opens.

2. Click the *Process Flow* tab.
3. From the directory where you saved dmorsumx, drag the flow into the workflow area.
4. Right-click the *Start* icon, drag the arrow to dmorsumx and release.
5. Right-click *dmorsumx* and click *Properties*.
6. Check the check box labeled *Get Parameters using Synonym*.

7. Click the ellipsis  button in the File Name entry area. In the ibisamp directory, select *dmplnts*.

   The properties for the flow should look like the following image.



8. Click *Save* on the Quick Access Toolbar to save the flow. Enter *dmorsumpx* as the flow name.
9. To create the target table, in the ibisamp directory located in the navigation pane, right-click the table named *dmorsum*, select *Data Management*, and then click *Recreate DBMS table*.
10. On the Flow tab, in the Run group, click *Run* and select *Submit*.

    When the flow completes, you should see a message in the console log:

```
DM: (ICM18763) Request ibisamp/dmorsumpx complete
```

Example: View Logs for a Parameterized Data Flow

When a process flow calls a data flow using the iterator, multiple logs are generated. This example shows how to view the logs.

1. With the dmorsumpx process flow open, on the Flow tab, in the Reports group, click *View Last Log*.

   The process flow log opens.

2. Click the blue line with the Job ID.

The iterator log opens. Note that the Parameters line show the values of each parameter for each run.

LOOPBACK: Log for ibisamp/dmorsumx

	Datetime	Message Code	Log Messages	Application	Name
1	2020/06/25 16:39:46	(ICM18972)	------ Start of Iterator Log Record for ibisamp/dmorsumx ------	ibisamp	dmorsumx
2	2020/06/25 16:39:46	(ICM18774)	Request ibisamp/dmorsumx created	ibisamp	dmorsumx
3	2020/06/25 16:39:46	(ICM18994)	Iterated job parameters: LCON='CON01 ',LTABLE='dmordbos',LPLANT='BOS'	ibisamp	dmorsumx
4	2020/06/25 16:39:46	(ICM18762)	Job ID: 20200625163946_f6a650f4	ibisamp	dmorsumx
5	2020/06/25 16:39:46	(ICM18774)	Request ibisamp/dmorsumx created	ibisamp	dmorsumx
6	2020/06/25 16:39:46	(ICM18994)	Iterated job parameters: LCON='CON01 ',LTABLE='dmorddal',LPLANT='DAL'	ibisamp	dmorsumx
7	2020/06/25 16:39:46	(ICM18762)	Job ID: 20200625163946_56bb77a3	ibisamp	dmorsumx
8	2020/06/25 16:39:46	(ICM18774)	Request ibisamp/dmorsumx created	ibisamp	dmorsumx
9	2020/06/25 16:39:46	(ICM18994)	Iterated job parameters: LCON='CON01 ',LTABLE='dmordorl',LPLANT='ORL'	ibisamp	dmorsumx
10	2020/06/25 16:39:46	(ICM18762)	Job ID: 20200625163946_e13582b3	ibisamp	dmorsumx
11	2020/06/25 16:39:46	(ICM18774)	Request ibisamp/dmorsumx created	ibisamp	dmorsumx
12	2020/06/25 16:39:46	(ICM18994)	Iterated job parameters: LCON='CON01 ',LTABLE='dmordstl',LPLANT='STL'	ibisamp	dmorsumx
13	2020/06/25 16:39:46	(ICM18762)	Job ID: 20200625163946_ac94fe48	ibisamp	dmorsumx
14	2020/06/25 16:39:46	(ICM18774)	Request ibisamp/dmorsumx created	ibisamp	dmorsumx
15	2020/06/25 16:39:46	(ICM18994)	Iterated job parameters: LCON='CON01 ',LTABLE='dmordla ',LPLANT='LA '	ibisamp	dmorsumx
16	2020/06/25 16:39:46	(ICM18762)	Job ID: 20200625163946_1372fcbf	ibisamp	dmorsumx
17	2020/06/25 16:39:46	(ICM18973)	------ End of Iterator Log Record for ibisamp/dmorsumx ------	ibisamp	dmorsumx

3. Click one of the blue lines with Job IDs. The log for an individual run of the data flow opens.

LOOPBACK: Log for ibisamp/dmorsumx

	Datetime	Message Code	Log Messages	Application	Name
1	2020/06/25 16:56:56	(ICM18974)	------ Start of Log Record for ibisamp/dmorsumx ------	ibisamp	dmorsumx
2	2020/06/25 16:56:56	(ICM18122)	Request - ibisamp/dmorsumx (Owner: SE02703) submitted.	ibisamp	dmorsumx
3	2020/06/25 16:56:56	(ICM18097)	DEP_0: Set variables	ibisamp	dmorsumx
4	2020/06/25 16:56:56	(ICM18742)	dmorsum type MS SQL Server OLE DB/AzureDB Existing target	ibisamp	dmorsumx
5	2020/06/25 16:56:56	(ICM18704)	Extended Bulk Load Utility	ibisamp	dmorsumx
6	2020/06/25 16:56:56	(ICM18743)	Starting Load	ibisamp	dmorsumx
7	2020/06/25 16:56:56		_EDATEMP/sqlin HELD AS SQL_SCRIPT	ibisamp	dmorsumx
8	2020/06/25 16:56:56	(FOC2644)	FULL OPTIMIZATION OF INSERT WITH SUBSELECT CANNOT BE DONE	ibisamp	dmorsumx
9	2020/06/25 16:56:56	(FOC2653)	SOURCE AND TARGET FILES HAVE DIFFERENT SUFFIXES MSODBC SQLMSS	ibisamp	dmorsumx
10	2020/06/25 16:56:56	(FOC2664)	PARAMETRIZED PER RECORD STATEMENT WILL BE GENERATED	ibisamp	dmorsumx
11	2020/06/25 16:56:56	(FOC2661)	TARGET FILE ibisamp/dmorsum	ibisamp	dmorsumx
12	2020/06/25 16:56:56	(FOC2662)	BULK APPEND PROCESS STARTED AT 16.56.56	ibisamp	dmorsumx
13	2020/06/25 16:56:56	(FOC2661)	TARGET FILE ibisamp/dmorsum	ibisamp	dmorsumx
14	2020/06/25 16:56:56	(FOC2663)	BULK APPEND PROCESS ENDED AT 16.56.56, ELAPSED TIME = 00:00:00.000	ibisamp	dmorsumx
15	2020/06/25 16:56:56	(FOC2661)	TARGET FILE ibisamp/dmorsum	ibisamp	dmorsumx
16	2020/06/25 16:56:56	(ICM18744)	Ending Load	ibisamp	dmorsumx
17	2020/06/25 16:56:56	(ICM18040)	Return Code = 0	ibisamp	dmorsumx
18	2020/06/25 16:56:56	(ICM18076)	Request: ibisamp/dmorsumx - finished processing	ibisamp	dmorsumx
19	2020/06/25 16:56:56	(ICM18007)	CPU Time : 47	ibisamp	dmorsumx
20	2020/06/25 16:56:56	(ICM18031)	Finished	ibisamp	dmorsumx
21	2020/06/25 16:56:56	(ICM18072)	Elapsed run time 0:00:00	ibisamp	dmorsumx
22	2020/06/25 16:56:56	(ICM18975)	------ End of Log Record for ibisamp/dmorsumx ------	ibisamp	dmorsumx

Using Subqueries

A subquery or subselect can be used in several different places in an SQL SELECT statement.

Example: Using a Subquery in the Select List

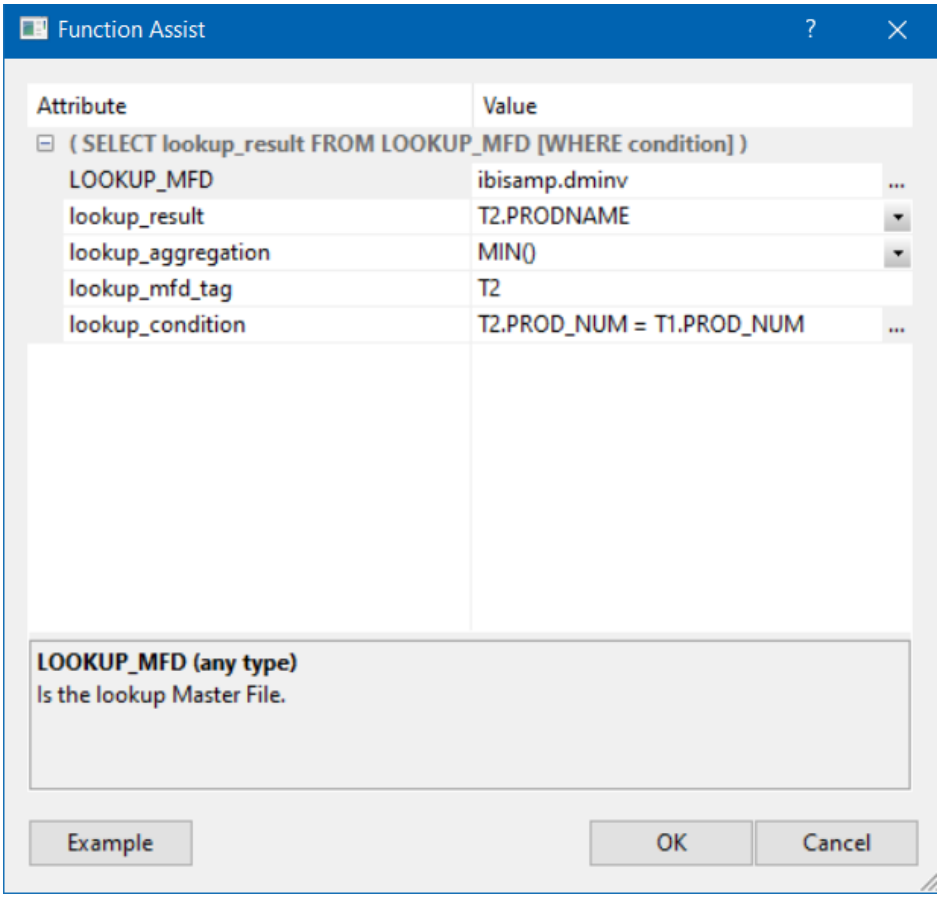
This example shows how subqueries can be used in the list of columns in a select statement to include values from additional tables without using a JOIN. The product name PRODNAME is retrieved from the inventory table dminv, and the store name STORENAME is retrieved from the table of the company dmcomp.

For the complete example, run the dmsubsel flow that is located in the ibisamp directory (ibisamp/dmsubsel).

1. In the DMC, right-click on an application directory in the navigation pane, point to *New*, and then click *Flow*.
2. From the ibisamp directory drag dmord as a source to the left of the SQL object.
3. Double-click the SQL object to open the Column Selection dialog box.
4. Ctrl + click *STORE_CODE, PROD_NUM, QUANTITY*, and *LINEPRICE* and click the *Move Columns Right* button to add them to the Selected Columns list.
5. Click the *Insert Columns* button.

   The SQL Calculator opens.
6. Enter PRODNAME for alias type.
7. Select the *Functions* tab and expand *SQL Operators*. Double-click the function that starts with (SELECT lookup_result).
8. Click the ellipsis for LOOKUP_MFD and look in ibisamp for the table dminv. Click *Select*.
9. Enter T2 for the lookup_mfd_tag type.
10. Click the ellipsis for lookup_result and click *PRODNAME*.
11. For lookup_aggregation, use the default MIN().
12. Click the ellipsis for lookup_condition. The condition calculator opens. Enter T2.PROD_NUM = T1.PROD_NUM.

The Function Assist dialog box opens, and should resemble the following image.



13. Click *OK* to close the Function Assist dialog box and click *OK* to close the SQL Calculator.
14. Repeat steps 5 to 14 using the following values:
    - For LOOKUP_MFD, enter DMCOMP.
    - For lookup_result, enter T3.STORENAME.
    - For lookup_aggregation, enter MIN().
    - For lookup_mfd_tag, enter T3.
    - For lookup_condition, enter T1.STORE_CODE = T3.STORE_CODE.
15. Click *OK* to close the Column Selection dialog.

16. From the navigation pane in the ibisamp directory, drag the synonym dmstore to the right of the SQL object.
17. Right-click *dmstores* and click *Properties*.

    The properties page opens.
18. For the Prior to load option, select *Truncate table* if available, or *Delete all rows* if not.

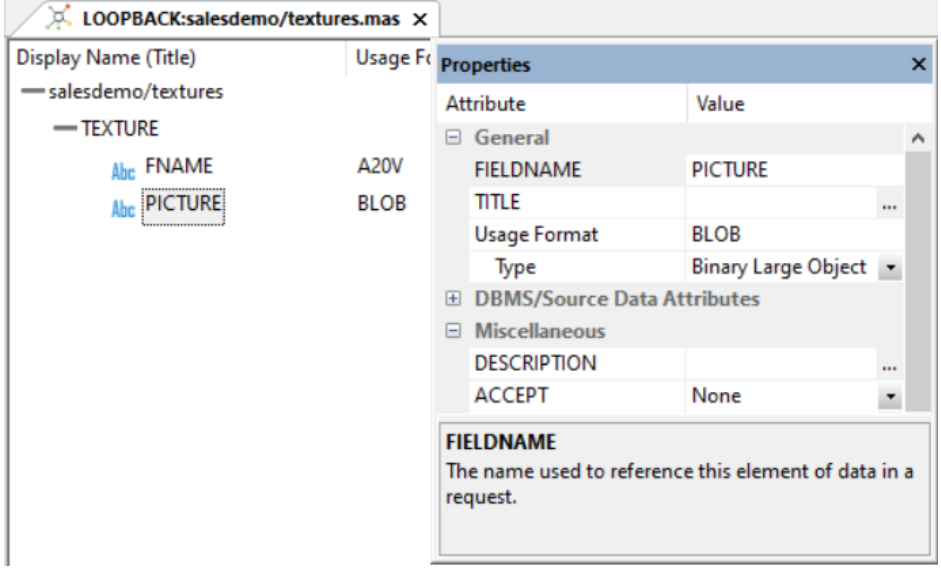
    Click the *X* to close the page.
19. Double-click *dmstores*.

    The Transformations window opens.
20. Click the *Automap* button. Next, click *OK* to close the Transformations window.
21. On the Flow tab, in the Run group, click *Run* and select *Submit* from the drop-down menu.
22. When the flow completes, right-click *dmstores*, select *Operations*, and then click *sample data*. Note that the STORENAME and PRODUCTNAME columns are filled in with values.

Loading Images Into a Database Table

Data Migrator has the capability to load image files into a database table by specifying their location in the input data.

The first step is to create a table to store the images. The table to load should have a column with a data type that can hold binary data. In the current release of MS SQL Server, that would be varbinary(max). In previous releases, it is image. In the synonym, the USAGE and ACTUAL fields for the column are set to BLOB (Binary Large OBject, as shown in the following image.



The following example outlines the remaining steps needed to load images into a database table.

Example: Loading Images Into a Database Table

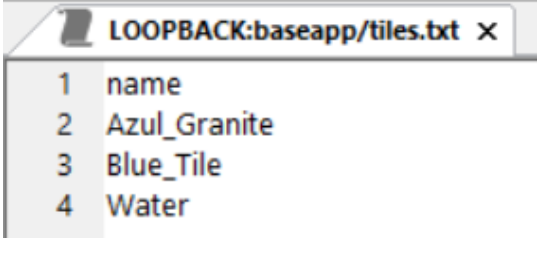
The first step is to map an application directory. The location of the files containing the images is specified in the usual appdir/filename format. To map the application directory:

1. Right-click *Application Directories*, select *New*, and then click *Application Directory*.

   The Create New Application dialog box opens.
2. From the Application Type drop-down menu, select *Application Mapping to Disk*.
3. In the Application Name box, enter *graph*.
4. In the Physical Location box, enter c:\ibi\srv82\home\graph.
5. From the Profile drop-down menu, select the desired profile.
6. Click *OK*.

   Next, create a source file and synonym. The source data needs a list of the names of the image files. From the DMC:
7. Right-click *baseapp*.
8. Select *New* and then click *File*.
9. Enter *names* as a header line, followed by the names of some of the .gif files (without the extension).
10. Click the *Save* button.
11. Enter *tiles* as the name and click *OK*.

    The following image shows the result.



    Then, create a synonym for the file.
12. Right-click *baseapp*, select *New*, and then click *Synonym*.
13. Right-click the node for Delimited Files, and click *Show local files*.

14. For Data File, select baseapp/tiles.txt.
15. Click *OK*.
16. For Header, select *Yes*, and click the Tiles synonym.
17. Click *Add*
18. Click *Close*.

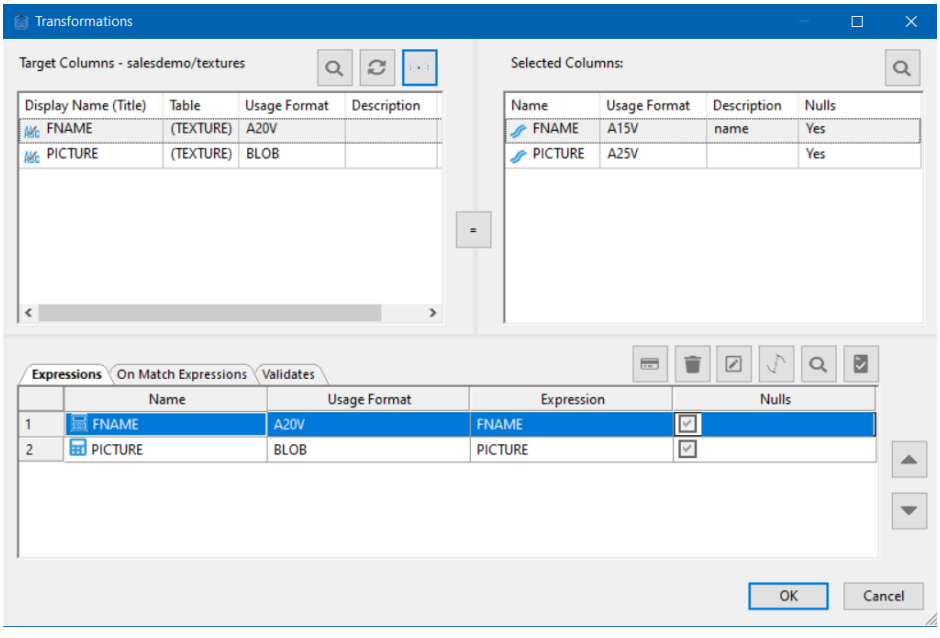
    Next, create a data flow that loads the images into a database table. The source is the delimited file tiles that contains the names of the image files. The target is the existing database table textures.

19. Right-click *baseapp*. Select *New* and then click *Flow*.
20. Right-click the left side of the workspace, select *Add Source*, and then click *baseapp/tiles*.
21. Right-click the right side of the workspace, select *Add Target*, then *Existing*, and then click *baseapp/textures*.
22. Double-click the *SQL* object to open it.
23. Double-click *Name* to add it to selected columns. Enter *FNAME* for SQL Alias.
24. Click the *Insert Columns* button.
25. For Alias, enter *PICTURE*. For Expression, enter *'graph/'* || *T1.NAME* || *'.gif'*.
26. Click the *Test SQL* button to see the values for the columns.
27. Close the Test SQL dialog box and click *OK* to close the SQL object.
28. Click *OK* to close the Column Selection dialog box.
29. Double-click the Target object.

    The Transformations window opens.

30. Click the *Automap* button.

The Transformations window is shown in the following image.



31. Click *OK* to close the Transformations window.
32. On the toolbar, click the *Run* button and click *Submit*.
33. Enter *loadim* for the procedure name and click *Save*.

    Wait for the completed message to appear in the console log and then click *View Last Log* to verify that the three rows were loaded.

    To see the images that were loaded into the table, you need to write a FOCUS report that sends its output to a PDF file.

34. RIght-click *baseapp*, select *New*, and then click *Procedure*.
35. Enter the following lines:

```
TABLE FILE textures
PRINT PICTURE
BY FNAME
ON TABLE PCHOLD AS baseapp/texturpt FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT,$
TYPE=DATA,COLUMN=PICTURE,IMAGE=(PICTURE),SIZE=(1 1),POSITION=(+0.5 +0.5),
$
END
```

36. Click *Run*.

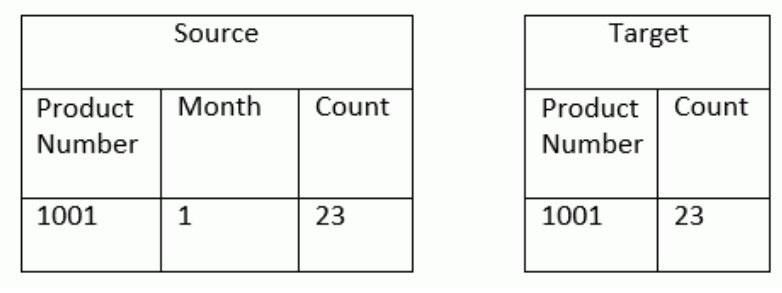
The PDF file will open in the workspace.

Updating Values in a Table

Normally, when using the Insert/Update load type when a row in the target table is updated, all of the values in the row of the target table except for the key column(s) are replaced with the new values.

However, there are circumstances where it is useful to *add* an incoming value to the value in the target table. For example, if the source table contains a count of units sold in some time period and you want the target table to contain the total count.

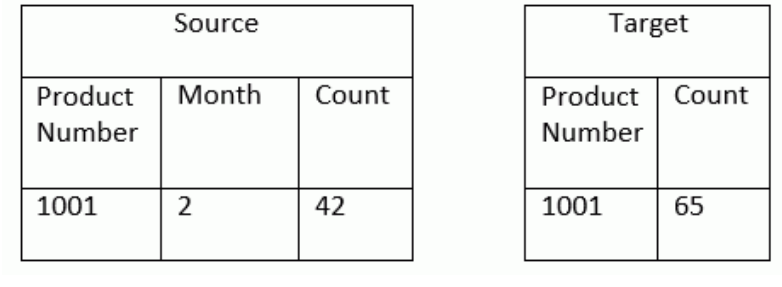
Consider a source file with one row, which is copied to a new target table, as shown in the following image.



Source		
Product Number	Month	Count
1001	1	23

Target	
Product Number	Count
1001	23

Now consider a new source file with one row, and instead of *replacing* the existing count with the new count, the new count is *added* to the existing count, as shown in the following image.



Source		
Product Number	Month	Count
1001	2	42

Target	
Product Number	Count
1001	65

The following examples will walk you through the steps of creating a synonym for a source table, writing a flow to create and load one month of data to a target table, and then writing another flow to add a subsequent month of data to the target table.

Example: Creating a Synonym for a CSV (Comma Separated Values) File

The source table for this tutorial can be found in the application directory where you created the tutorial files. By default, this location is the ibisamp directory.

From the DMC:

1. Right-click *ibisamp*, point to *New*, and click *Synonym (Create or Update)*.

   The Get Data dialog box opens.

2. Right-click the Delimited Files under Server Data Sources, and select *Show local files*.

   The Select Data File dialog box opens.

3. In the Data Files Location box, select *ibisamp*.
4. Click *sales_orders.csv*.
5. Click *OK*.

   The Create Synonym dialog box opens.

6. Select the checkbox next to the *sales_orders* synonym.

   To view the CSV file, double-click *sales_orders*

7. Set the following parameters to create the synonym:
   a. In the Field Delimiter, leave comma selected.
   b. In the Field Enclosure box, select *" (double quote)* from the drop-down menu.
   c. Select the Header row checkbox.
   d. Expand the Customize Data type mappings section. In the Decompose Data fields into components box, select *OFF - Turn Off* from the drop-down menu.
   e. In the Prefix box, enter *dm*.
   f. Select the *sales_orders* check box.
8. Click *Add*.

   The Status dialog box opens.

9. Ensure that the status shows "Created successfully" and no other messages. Click *Close*.
10. To view the data through the synonym, expand the *ibisamp* directory, right-click *dmsales_orders*, and click *Sample Data*.

Example: Creating an Initial Load Data Flow

The next step is to create a Data Flow that creates a new target table and loads the first month of data. In this example, we will be using *mysamp* as the application directory.

Right-click the Data Flow workspace and deselect *Optimize Load*.

From the DMC:

1. Right-click *mysamp*, point to *New*, and click *Flow*.

   The Data Flow workspace opens.

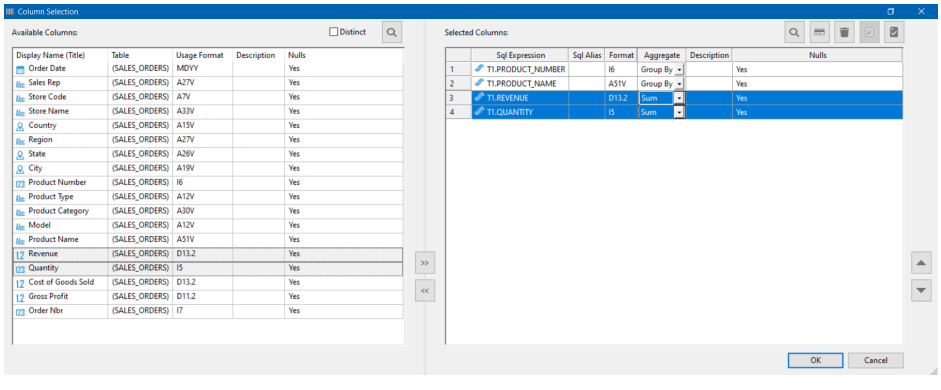
2. Right-click the left hand side of the workspace and click *Add Source*.

   The Select Synonym dialog box opens.

3. Click the *ibisamp* directory and then click *dmsales_orders*. Click *Select*.

   The source synonym is added to the workspace.

4. Double-click the SQL object to open it.
5. Hold the Ctrl key to select and select the *PRODUCT_NUMBER* and *PRODUCT_NAME* columns. Click the >> button to add them to the Selected Columns section.
6. With both columns selected, select *Group By* from the Aggregates drop-down menu.
7. From the Available Columns section, hold the Ctrl key and select the *REVENUE* and *QUANTITY* columns. Click the >> button to add them to the Selected Columns section.
8. With both columns selected, select *Sum* from the Aggregates drop-down menu, as shown in the following image.



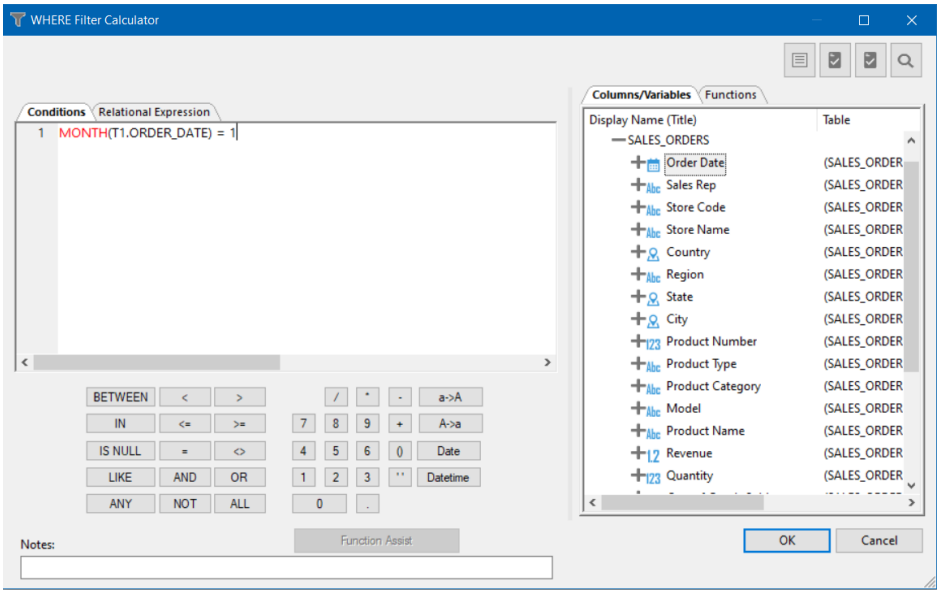
9. Click *OK* to close the Column Selection dialog box.
10. Right-click the SQL object and click *WHERE Filter*.

    The WHERE Filter calculator opens.

11. Click the *Conditions* tab.
12. Right-click the *ORDER_DATE* column, point to *Insert Function*, and click *MONTH* from the list of functions.

    The expression MONTH(T1.ORDER_DATE) is added.

13. Enter *=1* at the end of the expression, as shown in the following image.



14. Click *OK* to close the WHERE Filter calculator.
15. Right-click the right side of the workspace, point to *Add Target*, and click *New*.
16. Right-click the target object and click *Properties*.

    The Properties panel for the target opens.

17. Enter the following values:

    a. For Adapter, select any adapter for a relational database.

    b. For Synonym, enter the name of your application directory and a name for the target table For example, mysamp/dmsales_counts.

    c. For Load Type, use Insert/Update.

    d. In the Key box, under the Adapter Specific section, enter or select *PRODUCT_NUMBER*.

18. On the Flow tab, from the Run group, click *Run* and then click *Submit*.
19. Click *OK* in the Save changes dialog box.

    The Save Procedure As dialog box opens.

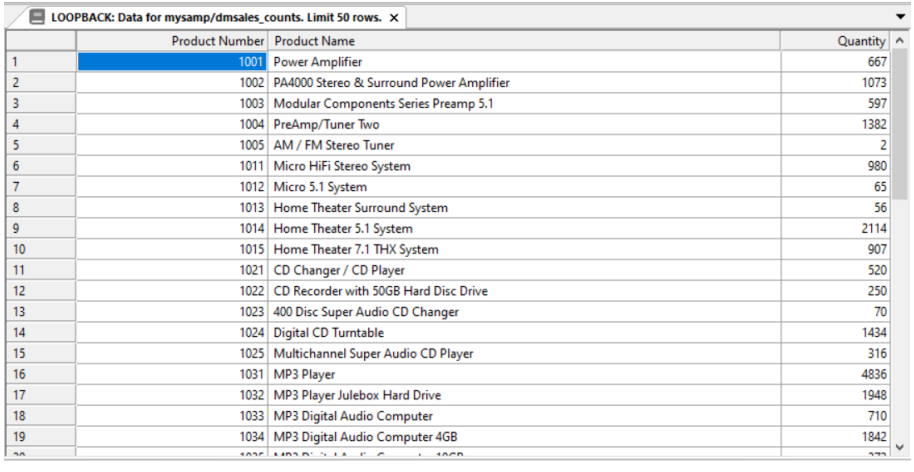
20. In the Name box, enter *create_dmsales_counts*.
21. Click *Save*.

    You should see a message in the Console Log at the bottom of the DMC that says Request mysamp/create_dmsales_counts complete.

Note: If the Console Log is not visible, on the Home tab, from the View group, click *Console Log*.

22. On the Flow tab, from the Reports group, click *View Last Log* to view the number of rows that were inserted. Click the *X* on the tab to close the log view and return to the flow.

23. To view the data that was loaded, right-click the target object, point to *Operations*, and click *Sample Data*. The report shows the product information that was loaded, specifically, the quantity of each. For example, Product Number 1005 shows a Quantity of 2, as shown in the following image.



LOOPBACK: Data for mysamp/dmsales_counts. Limit 50 rows.

	Product Number	Product Name	Quantity
1	1001	Power Amplifier	667
2	1002	PA4000 Stereo & Surround Power Amplifier	1073
3	1003	Modular Components Series Preamp 5.1	597
4	1004	PreAmp/Tuner Two	1382
5	1005	AM / FM Stereo Tuner	2
6	1011	Micro HiFi Stereo System	980
7	1012	Micro 5.1 System	65
8	1013	Home Theater Surround System	56
9	1014	Home Theater 5.1 System	2114
10	1015	Home Theater 7.1 THX System	907
11	1021	CD Changer / CD Player	520
12	1022	CD Recorder with 50GB Hard Disc Drive	250
13	1023	400 Disc Super Audio CD Changer	70
14	1024	Digital CD Turntable	1434
15	1025	Multichannel Super Audio CD Player	316
16	1031	MP3 Player	4836
17	1032	MP3 Player Julebox Hard Drive	1948
18	1033	MP3 Digital Audio Computer	710
19	1034	MP3 Digital Audio Computer 4GB	1842

Example: Creating an Update Data Flow

In this step, we will start with the initial load flow and edit it so that it updates values in the target table by adding new to existing values.

Right-click the flow canvas, select *Flow Properties*, and deselect *Optimize Load*. Then close the Properties panel.

1. In the Data Flow that was just created, right-click the SQL object and click *WHERE Filter*.

   The WHERE Filter calculator opens.

2. Click the *Conditions* tab.
3. In the expression, change the number 1 to a *2*.
4. Click *OK*.
5. Right-click the target object and click *Delete*.
6. Click *Yes* to confirm that you want to delete the target object.

7. Right-click the right hand side of the workspace, point to *Add Target*, and click *Existing*.

   The Select Synonym dialog box opens.

8. Select the application directory you used to create the target table. In this example, it is *mysamp*. Then, select the target table in the available list, for example, *mysamp/dmsales_counts*.

9. Click *Select*.

10. Right-click the new target object and click *Properties*.

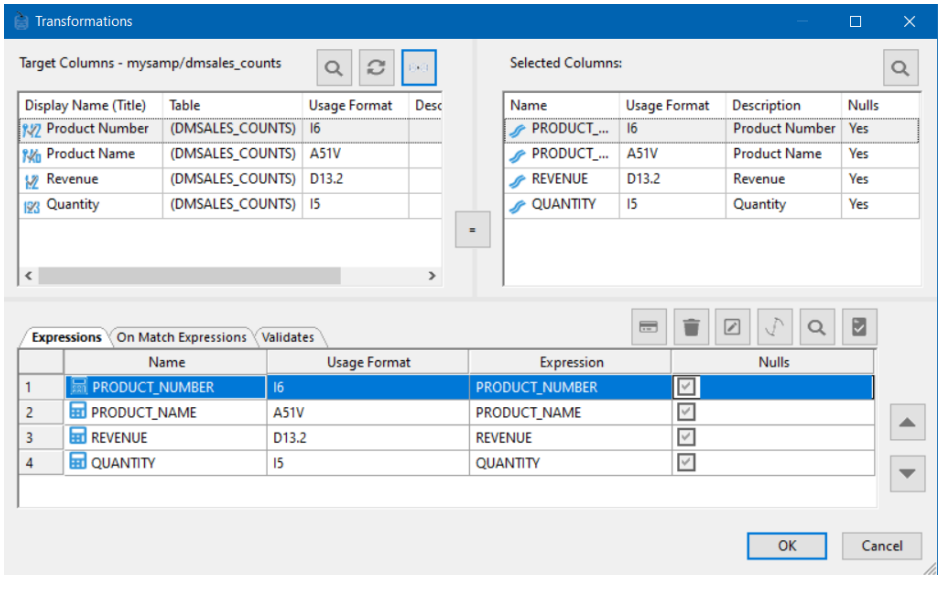
    The Properties pane opens.

11. Select *Update the existing record* from the *If the record exists* drop-down menu.

12. Double-click the target object.

    The Transformations dialog box opens.

13. Click the AutoMap button to automatically map the source to the target columns. This defines the assignments that are done from source to target when inserting a new row, as shown in the following image.



14. Click the *On Match Expressions* tab.

15. In the Target Columns section, double-click *PRODUCT_NUMBER*.

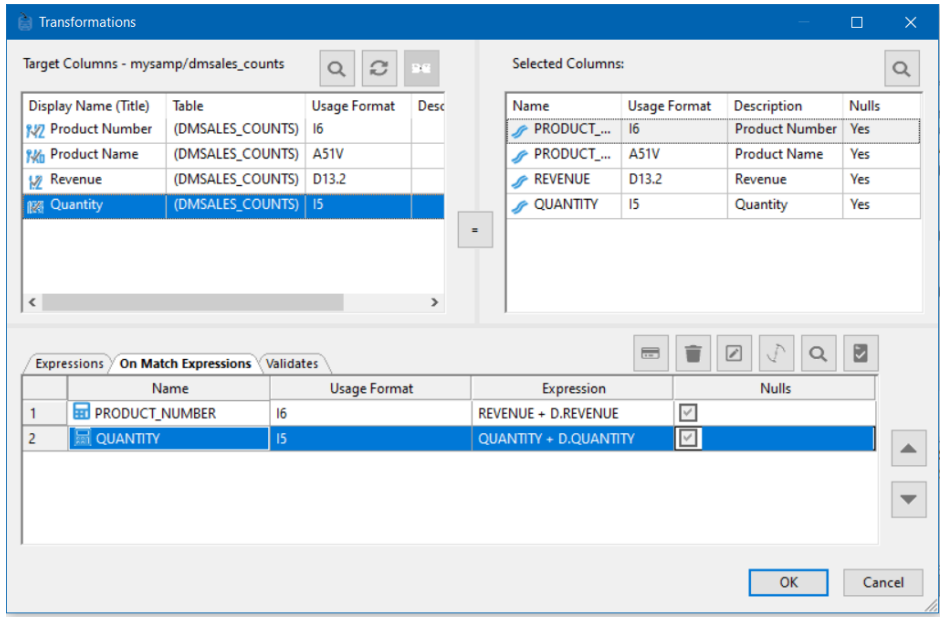
    The On Match Transformation calculator opens.

16. Double-click *REVENUE* to add it to the expression. This is the value from the source table.

17. Click the plus sign.
18. Expand *dmsales_counts* and double-click *REVENUE*. This will add D.REVENUE to the expression, which is the value from the *target* table.
19. Click *OK* to close the On Match Transformation calculator.
20. In the Target Columns section of the Transformations dialog box, double-click *QUANTITY* .

    The On Match Transformation calculator opens.
21. Double click *QUANTITY* to add it to the expression.
22. Click the plus sign.
23. Expand dmsales_counts and double click *QUANTITY*. This will add D.QUANTITY to the expression, which is the value from the *target* table.
24. Click *OK* to close the On Match Transformation calculator.

    The Transformations dialog box should look like the following image.



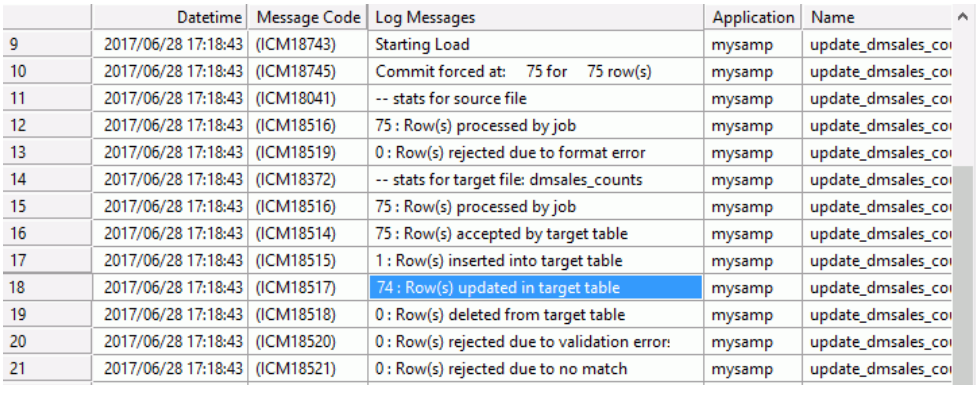
25. Click *OK* to close the Transformations dialog box.
26. On the Quick Access Toolbar, click the *Save As* button.

    The Save Procedure As dialog box opens.
27. Enter a name for the procedure, for example, *update_dmsales_counts*.
28. Click *Save*.

29. On the Flow tab, from the Run group, click *Run* and then click *Submit*.

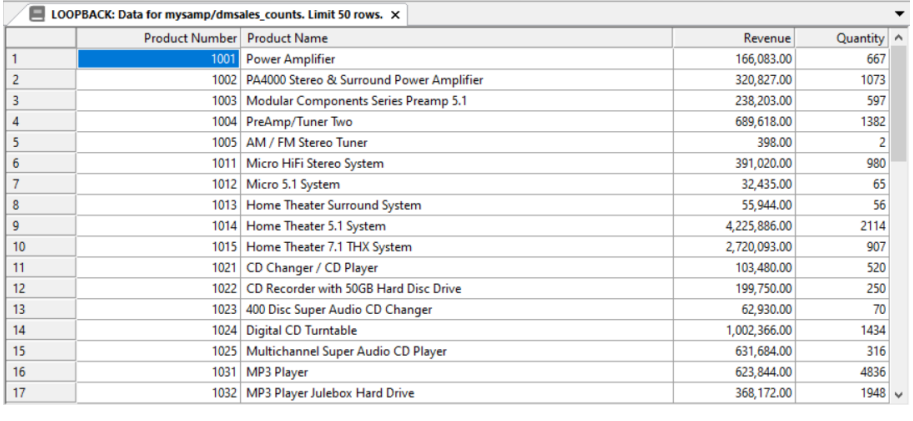
    The message "(ICM18763) Request ibisamp/update_dmsales_counts complete" should appear in the console log, confirming that the flow was submitted successfully.

30. On the Flow tab, from the Reports group, click *View Last Log* to view the log. The log shows that 1 row was inserted, and 74 were updated, as shown in the following image.



	Datetime	Message Code	Log Messages	Application	Name
9	2017/06/28 17:18:43	(ICM18743)	Starting Load	mysamp	update_dmsales_co
10	2017/06/28 17:18:43	(ICM18745)	Commit forced at: 75 for 75 row(s)	mysamp	update_dmsales_co
11	2017/06/28 17:18:43	(ICM18041)	-- stats for source file	mysamp	update_dmsales_co
12	2017/06/28 17:18:43	(ICM18516)	75 : Row(s) processed by job	mysamp	update_dmsales_co
13	2017/06/28 17:18:43	(ICM18519)	0 : Row(s) rejected due to format error	mysamp	update_dmsales_co
14	2017/06/28 17:18:43	(ICM18372)	-- stats for target file: dmsales_counts	mysamp	update_dmsales_co
15	2017/06/28 17:18:43	(ICM18516)	75 : Row(s) processed by job	mysamp	update_dmsales_co
16	2017/06/28 17:18:43	(ICM18514)	75 : Row(s) accepted by target table	mysamp	update_dmsales_co
17	2017/06/28 17:18:43	(ICM18515)	1 : Row(s) inserted into target table	mysamp	update_dmsales_co
18	2017/06/28 17:18:43	(ICM18517)	74 : Row(s) updated in target table	mysamp	update_dmsales_co
19	2017/06/28 17:18:43	(ICM18518)	0 : Row(s) deleted from target table	mysamp	update_dmsales_co
20	2017/06/28 17:18:43	(ICM18520)	0 : Row(s) rejected due to validation error:	mysamp	update_dmsales_co
21	2017/06/28 17:18:43	(ICM18521)	0 : Row(s) rejected due to no match	mysamp	update_dmsales_co

31. To view the data that was loaded, return to the flow, right-click the target object, point to *Operations*, and click *Sample Data*. The report shows that product 1005 now has a Quantity of 445, as shown in the following image.



LOOPBACK: Data for mysamp/dmsales_counts. Limit 50 rows.

	Product Number	Product Name	Revenue	Quantity
1	1001	Power Amplifier	166,083.00	667
2	1002	PA4000 Stereo & Surround Power Amplifier	320,827.00	1073
3	1003	Modular Components Series Preamp 5.1	238,203.00	597
4	1004	PreAmp/Tuner Two	689,618.00	1382
5	1005	AM / FM Stereo Tuner	398.00	2
6	1011	Micro HiFi Stereo System	391,020.00	980
7	1012	Micro 5.1 System	32,435.00	65
8	1013	Home Theater Surround System	55,944.00	56
9	1014	Home Theater 5.1 System	4,225,886.00	2114
10	1015	Home Theater 7.1 THX System	2,720,093.00	907
11	1021	CD Changer / CD Player	103,480.00	520
12	1022	CD Recorder with 50GB Hard Disc Drive	199,750.00	250
13	1023	400 Disc Super Audio CD Changer	62,930.00	70
14	1024	Digital CD Turntable	1,002,366.00	1434
15	1025	Multichannel Super Audio CD Player	631,684.00	316
16	1031	MP3 Player	623,844.00	4836
17	1032	MP3 Player Julebox Hard Drive	368,172.00	1948

Chapter 12

Change Data Capture

Change Data Capture (CDC) enables reading from database logs to determine what rows in a table have changed. This approach is useful when loading from large source tables where it would take too long to process the entire table.

Databases currently supported are MariaDB 10, MySQL 5.7 and higher, Microsoft SQL Server 2008, 2012, and 2014 Enterprise Edition, PostgreSQL version 12 and higher, UDB/Db2, and ORACLE 10g, 11g, and 12c Enterprise Edition. Additional databases will be supported in future releases.

To use Change Data Capture, the database must be configured to use database logging. In most organizations this must be done by a database administrator. A Table Log Records synonym is created for a log table, allowing reading from the database log records for a selected table. This CDC synonym can be used as a source in a data flow or in a direct load flow. The target synonym on such flows has a load type of Change Data Capture

In this chapter:

- Enabling the Database for Logging in Relational Databases
- Properties of a Table Log Record Synonym
- Tutorial: Using Change Data Capture with Relational Databases

Enabling the Database for Logging in Relational Databases

To use Change Data Capture, the source database must be configured to use database logging. In most organizations this must be done by a database administrator.

This example uses *dminv* as the source table. To create this table, see *How to Create Sample Procedures and Data* on page 59.

The database table must be enabled for logging, and the procedure depends on the database being used.

Configuring CDC in Microsoft SQL Server®

Microsoft SQL Server supports Change Data Capture in the Enterprise, Evaluation, and Developer editions.

To use logging, the source database must be configured to use database logging. In most organizations, this must be done by a database administrator.

Logging must be enabled at the database level since it is disabled by default. The database table must be enabled for logging as well. To enable logging, you must be connected as a member of the sysadmin server role.

To view the logging on a database level, connect to the database you want to enable and issue the following commands:

```
exec sys.sp_cdc_enable_db
select name, is_cdc_enabled from sys.databases
```

You will see a list of all databases and a 1 if logging is enabled for the database, or a 0 otherwise.

To check if the SQL Server Agent is running, open the Microsoft SQL Server Management Studio and navigate to the Object Explorer window. Click the icon for SQL Server Agent. It should have a green arrow. If it does not, right-click *SQL Server Agent* and click *Start*.

The SQL Server Agent and two jobs must be running. The jobs are created automatically when the database is enabled. For example, if the database is called *main*, the jobs would be named *cdc.main_capture* and *cdc.main_cleanup*.

After the SQL Server job Cleanup completes, the content of the current SQL Server transaction log changes. If the checkpoint file created by Data Migrator contains the LSN value, which is not within a Validity Interval of the retained transaction log, the Adapter setting LOG_SWITCHED is taken into account.

The Adapter setting LOG_SWITCHED is located on the Change Setting screen:

```
ENGINE engine SET LOG_SWITCHED { STOP | PROCEED }
```

The default value is STOP.

For more information on the checkpoint file, see *Creating a Data Migrator Direct Load Flow* on page 663.

In some situations, the updates on the database occur and the Cleanup job runs, but the Data Migrator flow is not running after the Cleanup job. In this case, the Validity Interval on the database might not have the LSN value that is matching the LSN in the checkpoint file.

When the value of the LOG_SWITCHED is set to STOP and the LSN value in the checkpoint file is not within a Validity Interval of the retained transaction log, the CDC flow returns the following error:

```
(FOC45008) INVALID CHECKPOINT
```

The checkpoint file is not updated at this point.

Note: To continue the successful flow execution, delete the existing checkpoint file.

When the value of the LOG_SWITCHED setting is set to PROCEED and the LSN value in the checkpoint file is not within a Validity Interval on the database, the CDC flow proceeds and updates the checkpoint file.

Note: In this situation, the change data values could be not updated by a flow as missing in the Validity Interval.

Each table to be used must have CDC enabled. This can be done from Microsoft SQL Server Management Studio. You must be a member of the db_owner database role to run the stored procedure. Issue the following commands, where *tablename* is the name of the table.

```
exec sys.sp_cdc_enable_table @source_schema = 'dbo', @source_name =
'tablename', @role_name = NULL
select is_tracked_by_cdc FROM sys.tables WHERE name = 'tablename'
```

Example:

```
exec sys.sp_cdc_enable_table @source_schema = 'dbo', @source_name =
'dminv', @role_name = NULL
select is_tracked_by_cdc FROM sys.tables WHERE name = 'dminv'
```

You will see a 1 if the table is enabled, and a 0 otherwise.

Configuring CDC in MySQL

MySQL supports Change Data Capture.

Preparing the Environment for CDC with MySQL®

In order to use CDC with MySQL, you must install Java® and MySQL client utilities on the system where the Server is installed. Additionally, on that system, you must have the Environment Variable CLASSPATH pointing to the location where the Connector/J is installed. Note that the location of Connector/J itself can be anywhere on your system. For complete documentation on configuring data adapters for MySQL, see the *TIBCO WebFOCUS® Adapter Administration* manual.

Java®

To use CDC with the MySQL adapter, you must have Java installed. The system Environment Variables JAVA_HOME or JDK_HOME must be specified, and the location of Java must be added to the PATH Environment Variable.

For example, if your Server is on Linux, add the following lines to your profile:

```
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0_91
export JDK_HOME=/usr/lib/jvm/jdk-1.8.0_91
export PATH=$JAVA_HOME/bin:$PATH
```

Connector/J

Connector/J is an interface that provides MySQL support for Java client programs that use JDBC connections. Connector/J is a JAR file with a standard name of mysql-connector-java-*version*-bin.jar. The file can be installed on any system on your environment.

The location of the Connector/J must be specified in either the Environment Variable CLASSPATH on the same system where Data Migrator is installed, or in a IBI_CLASSPATH variable of a Reporting Server. To specify the location of the Connector/J in the IBI_CLASSPATH variable of a Reporting Server, perform the following steps:

1. In the DMC, expand *Workspace* and expand the *Java Services* folder.
2. Right-click *DEFAULT* and select *Properties.*

   The Java Services Configuration pane opens.
3. Expand the *Class path* section.
4. In the IBI_CLASSPATH field, specify the full path file name to the MySQL Connector J JAR file.
5. Click *Save and Restart Java Services*.

Utilities for Processing Binary Log Files

While the MySQL database does not have to be installed on the same system where the Reporting server is installed, the MySQL client utilities for processing binary log files, *mysql* and *mysqlbinlog*, must be placed into a directory on the same system where Data Migrator is installed.

The version of these MySQL client utilities must be the same as the version of the MySQL database. The Environmental Variable PATH should list the path to the directory where the MySQL client utilities are placed.

The maximum size of the binary log files is set in the max_binlog_size system variable, and has a maximum possible value of 1 GB. The Binary Log is flushed when its size reaches its maximum, 1 GB. As a general rule, avoid using the LONGTEXT data type in a table that is used for CDC operations, as this data type has large storage requirements.

Configuring MySQL

MySQL enables logging at the database level by storing the history of database changes in the Binary Log Files. These Binary Log Files are used by Data Migrator for CDC operations and must use row-based logging. For example:

```
--binlog-format=ROW
```

Although it is possible to change the logging format at runtime, the best practice is to set it at a database server startup.

A user ID assigned on a connection to a MySQL database must have either a SUPER privilege or three global privileges: REPLICATION CLIENT, REPLICATION SLAVE, and SELECT. This task is typically performed by a database administrator. One method to assign such privileges is to issue the SQL statements shown in the following examples:

To create *user1* on any host:

```
CREATE USER 'user1'@'%' IDENTIFIED BY ‘password1’;
```

To grant *user1* all privileges:

```
GRANT ALL PRIVILEGES ON *.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

To enable *user1* for reading the binary logs from any subordinate host:

```
GRANT REPLICATION CLIENT ON *.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

To enable *user1* for reading the binary logs from any master host:

```
GRANT REPLICATION SLAVE ON *.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

To enable *user1* for selecting all objects in a database named *dbname1*:

```
GRANT SELECT ON dbname1.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

Configuring CDC in Oracle®

Configuring ORACLE for Change Data Capture can be done from Oracle SQL Developer or SQL Plus. In most organizations, this must be done by a database administrator.

```
connect sys/* as sysdba
```

To use the Change Data Capture, the database must be configured to use archiving. Enter the command:

```
ALTER SYSTEM ARCHIVE LOG START;
```

When using ORACLE XE (Express Edition), enter the following commands instead:

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
```

Confirm that archive log is enabled with the command:

```
select log_mode from v$database;
```

You should see the following result:

```
LOG_MODE
------------
ARCHIVELOG
```

To retrieve log information, a user ID needs to access certain system tables through public views. Issue these commands, where *userid* represents the user ID running CDC jobs.

```
grant select on "SYS"."V_$LOG" to userid ;
grant select on "SYS"."V_$LOGMNR_CONTENTS" to userid ;
grant select on "SYS"."V_$DATABASE" to userid;
```

To enable logging, a user ID must be granted the following, where *userid* represents the user ID to be used.

```
GRANT EXECUTE ON dbms_logmnr TO userid;
GRANT SELECT ANY TRANSACTION TO userid;
```

The table being logged must have all data columns logged by issuing the following command, where *tablename* represents the table to be logged.

```
ALTER TABLE tablename ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

Add supplemental log information for the database by entering:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

You can confirm that supplemental logging is enabled for a table with the following command. It should show YES.

```
select distinct
b.owner , b.object_name, supplemental_log_data_min
from v$database a, sys.all_objects b
where b.owner = 'userid'  and  b.object_name = 'tablename'
```

Note: If you recreate a database table, you must re-issue this command.

Change Data Capture for Oracle uses the DBMS_LOGMNR package to extract values from the Oracle redo log files. This does not support the following data types:

- LONG
- LONG RAW
- CLOB
- BLOB
- NCLOB
- ADT
- COLLECTION

Configuring CDC in PostgreSQL®

Data Migrator can support CDC with PostgreSQL through the consumption of Logical decoding in PostgreSQL.

Preparing the Environment for CDC With PostgreSQL

In order to use CDC with PostgreSQL, you must install the following on the system where the Reporting Server is installed.

- Java.
- PostgreSQL.
- The JDBC Driver.
- The output plugin wal2json.

Reference: **Configuring Java and the JDBC Driver**

To use CDC with the Adapter for PostgreSQL, you must have Java installed. The System Environment Variable JAVA_HOME or JDK_HOME must be specified, and the JAVA_HOME location must be added to the PATH Environment Variable.

For example, if your Reporting Server is on Linux, add the following lines to your profile:

```
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0_91
export JDK_HOME=/usr/lib/jvm/jdk-1.8.0_91
export PATH=$JAVA_HOME/bin:$PATH
```

The JDBC Driver jar file is needed in order to use CDC with PostgreSQL. You should download the jar file and save it to your path since you will be entering it in the IBI_CLASSPATH entry field during the configuration process. You can download the JAR file from the following site:

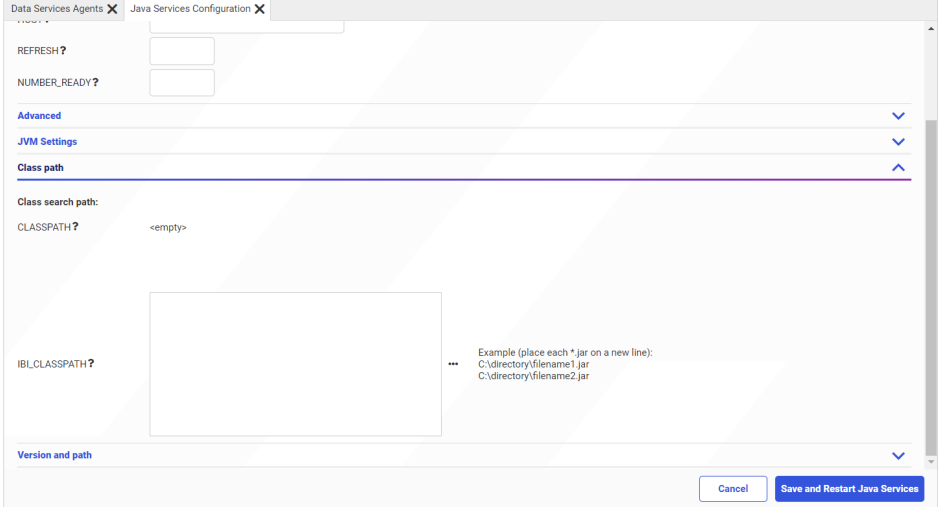
https://www.postgresql.org/download/

Copy the JDBC Driver jar file to any location on your system.

To specify the location of the jar file in the IBI_CLASSPATH variable of a Reporting Server, perform the following steps.

1. In the Web Console, navigate to the Workspace page or, in the Data Management Console (DMC), expand Workspace. Then expand the Java Services folder.
2. Right-click *DEFAULT* and click *Properties*.

   The Java Services Configuration pane opens.
3. Expand the Class path section, as shown in the following image.



4. In the IBI_CLASSPATH field, enter the full path file name to the JDBC Driver jar file.
5. Click *Save and Restart Java Services*.

Reference: Configuring Use of Write-Ahead Logs

Logical decoding in PostgreSQL uses an output plugin called *wal2json*. This plugin converts the PostgreSQL Write-Ahead Log (WAL) to a readable format.

The plugin is available for download from the site *https://github.com/eulerto/wal2json*.

Follow the instructions on the site to install the plugin.

The following is an example installation command for Red Hat Enterprise Linux (RHEL)/CentOS 7 or 8:

```
sudo yum install wal2json12
```

The following is an example installation command for Debian:

```
sudo dnf install wal2json12
```

The following is an example installation command for Ubuntu:

```
sudo apt-get install postgresql-12-wal2json
```

When using Windows, follow the instructions from the same site and build the wal2json.dll with the following considerations.

- Use non-debug mode.
- Use the same operating system where PostgreSQL is installed.
- During the build, target the same version of PostgreSQL as the version used with the DLL.

Copy the created wal2json.dll to the pkglibdir location. To find this location, open a command prompt and navigate to the PostgreSQL bin directory. Use the following command, where PGDATA is an environmental variable that points to the *data* directory.

```
cd %PGDATA%\..\bin
```

From this location, enter the following command.

```
pg_config --pkglibdir
```

The following is a sample path on Windows.

```
C:\Program Files\PostgreSQL\12
```

In order to use the write-ahead logs, you must edit the postgresql.conf configuration file and set the options for the PostgreSQL *wal_level* parameter, which determines the information that is written to the WAL. This configuration file is in the data area directory:

```
%PGDATA%/postgresql.conf
```

The wal_level parameter should be set to the value *logical*, in order to support logical decoding.

If you connect to the database as user *postgres*, you can edit the postgresql.conf file using the following SQL statement.

```
ALTER SYSTEM SET wal_level = logical;
```

After making the changes, you need to recycle the server as user *postgres*. For example:

```
%PGDATA%\..\bin pg_ctl restart
```

To verify the value of the wal_level parameter, you can issue the following SQL statement.

```
SHOW wal_level;
```

Using Logical Decoding and Monitoring Slots

Using a replication slot requires PostgreSQL replication privileges.

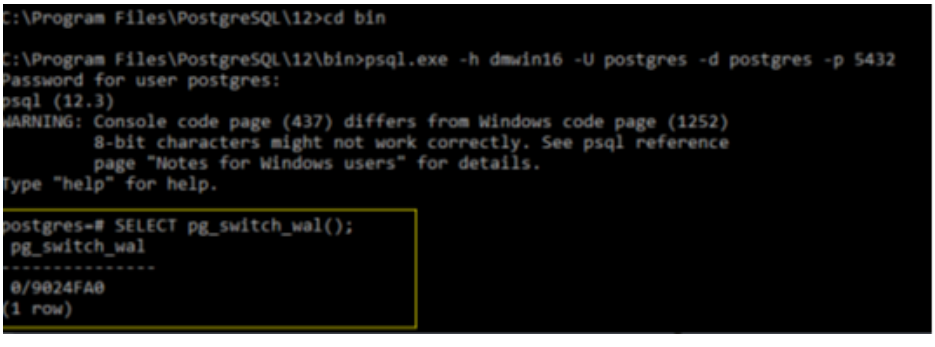
Each replication slot in PostgreSQL represents a stream of changes from a single database.

Procedure: **How to Create a Replication Slot**

1. Connect to the database with a user that has a SUPERUSER permissions.
2. To archive the current write-ahead log file and to switch to a new log file, enter the following command:

   ```
   SELECT pg_switch_wal();
   ```

   The following image shows issuing this command in Windows.



3. When a slot is no longer needed, drop it.

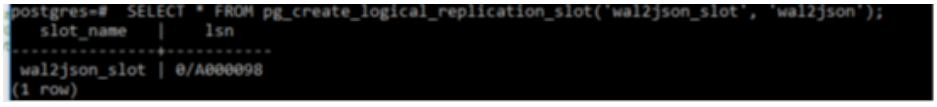
   According to the PostgreSQL documentation, the logical replication slots are consumed continuously, and you need to drop a slot as soon as the replication slot is not needed anymore. Use the following command:

   ```
   SELECT pg_drop_replication_slot('wal2json_slot');
   ```

4. Add new replication slots as needed, using the following command.

   ```
   SELECT * FROM pg_create_logical_replication_slot('wal2json_slot',
   'wal2json');
   ```

The following image shows issuing this command in Windows.



Reference: Monitoring Slots

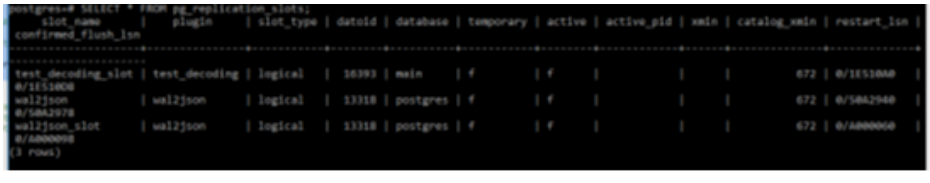
The following commands are used for logical decoding and monitoring the replication slots.

Listing Replication Slots

To list all replication slots in the primary server, issue the following command:

```
SELECT * FROM pg_replication_slots;
```

The following image shows issuing this command in Windows.

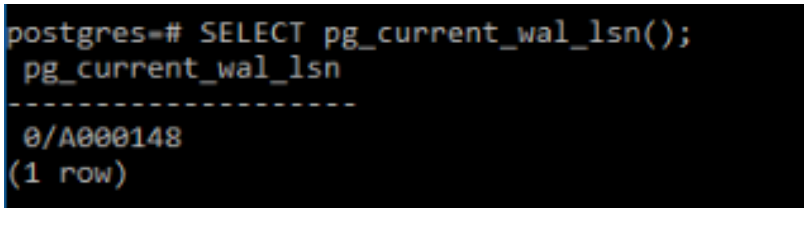


Listing the Current WAL Write Location

To list the current write-ahead log write location, issue the following command:

```
SELECT pg_current_wal_lsn();
```

The following image shows issuing this command in Windows.



Reference: Creating a User ID for Connection to PostgreSQL

A user ID on a connection to a PostgreSQL database must have the LOGIN and REPLICATION permissions. Please note that it is not advisable to add a SUPERUSER permission to this user, since it is insecure. This role allows you to bypass all permission checks. The following is sample syntax to create the user:

```
CREATE ROLE cdcuser1 WITH
    LOGIN
    NOSUPERUSER
    CREATEDB
    REPLICATION
    PASSWORD 'password';
```

Use the following statement to grant privileges to the user on the schema that will use the replications:

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA cdcschema1  TO cdcuser1;
```

Use the following statement to make the user an owner of the schema:

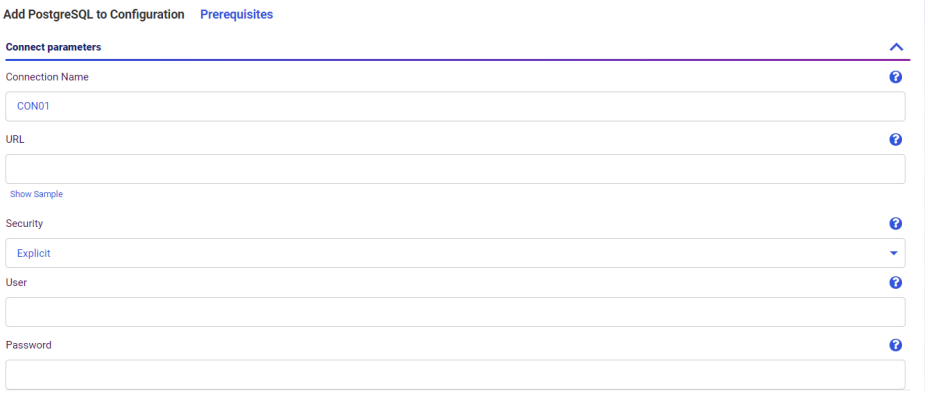
```
ALTER SCHEMA cdcschema1  OWNER TO cdcuser1;
```

Configuring the Adapter for PostgreSQL

You must create a connection for the Adapter for PostgreSQL that accesses the environment you prepared.

Reference: **Connection Attributes for the Adapter for PostgreSQL**

The following image shows the configuration parameters for the Adapter for PostgreSQL.



From your PostgreSQL DBA, obtain the values for the node name, port number and database name, then enter them in the URL entry field in the following format.

```
jdbc:postgresql://nodename:port/dbname
```

Leave Security as *Explicit*, and enter the User ID and Password associated with the role that you configured.

Configuring CDC in IBM Db2 Universal Database® on Windows and UNIX

Note that CDC support for UDB and Db2 requires the CLI interface. To enable the database for logging, issue the following commands from the Db2 command center:

```
UPDATE DATABASE CFG FOR database USING DFT_SQLMATHWARN YES
UPDATE DATABASE CFG FOR database USING LOGARCHMETH1 LOGRETAIN
UPDATE DBM CFG USING DISCOVER DISABLE
UPDATE DBM CFG USING DISCOVER_INST DISABLE
GRANT DBADM ON database TO USER userid
```

The Db2 table must have capture mode enabled. This can be done from the Db2 Command Center or a stored procedure.

In order to use Change Data Capture, each table must have data changes written to the log in an expanded format. Enter the following command, where *tablename* is the name of the table:

```
ALTER TABLE tablename DATA CAPTURE CHANGES
```

To return a table back to normal logging, use the following command:

```
ALTER TABLE tablename DATA CAPTURE NONE
```

Note: If you recreate a database table, you must re-enable expanded logging. To see which tables have expanded logging, you can issue the following Db2 select statement.

```
SELECT CREATOR, NAME, TYPE
FROM SYSIBM.SYSTABLES WHERE DATA_CAPTURE = 'Y'
```

Configuring CDC in IBM Db2 Universal Database on System Z

Note: CDC support for UDB and Db2 requires the CLI interface.

To enable the database for logging, issue the following Db2 commands:

```
UPDATE DATABASE CFG FOR database USING DFT_SQLMATHWARN YES
UPDATE DATABASE CFG FOR database USING LOGARCHMETH1 LOGRETAIN
UPDATE DBM CFG USING DISCOVER DISABLE
UPDATE DBM CFG USING DISCOVER_INST DISABLE
GRANT DBADM ON database TO USER userid
```

In order to use Change Data Capture, each table must have data changes written to the log in an expanded format. Enter the following command, where *tablename* is the name of the table:

```
ALTER TABLE tablename DATA CAPTURE CHANGES
```

To return a table back to normal logging, use the following command:

```
ALTER TABLE tablename DATA CAPTURE NONE
```

Note: If you recreate a database table, you must re-enable expanded logging. To see which tables have expanded logging, you can issue the following Db2 select statement:

```
SELECT CREATOR, NAME, TYPE
FROM SYSIBM.SYSTABLES WHERE DATACAPTURE = 'Y'
```

Configuring CDC in IBM Db2 Universal Database on iSeries

Note:

- The prerequisite requirement for CDC with Db2 on iSeries (IBM i) is to assign the value *MAXOPT3 to attribute RCVSIZOPT (receiver size options) at the initial stage, when the journal is created. Example commands are:

  ```
  CRTJRN JRN(LIBRARY1/JRNLB) JRNRCV(LIBRARY1/RCV01) RCVSIZOPT(*MAXOPT3
  *RMVINTENT)
  ```

  When the journal receiver is changed, the new receiver gets a new (LSN) sequence number. Assign the *CONT value to the SEQOPT attribute to allow your journal entry sequence number to continue in the subsequent receivers. For example:

  ```
  CHGJRN JRN(LIB71M1/QSQJRN) JRNRCV(*GEN) RCVSIZOPT(*RMVINTENT *MAXOPT3)
  SEQOPT(*CONT)
  ```

- The name of the journaling Db2 table on IBM i must not exceed 10 characters. To process DELETE records, make sure that the *BOTH (before-images and after-images) value has been set for the journal IMAGES parameter.

- **Note:** Your flow retrieves the journal entries that are presently available, regardless of the journal state. To ensure that your flow gets the most current updates from the journal entries, verify that your journal is in the *ACTIVE state.

The following setting for the Db2 adapter on IBM i allows processing of the 'before-images' (UB type) journal records (additionally to the 'after-images'), and helps with the updates of the target table of the flow if the keys fields on a source table had been changed:

```
ENGINE DB2 CDC_UPDATE_BEFORE_IMAGE {OFF | ON}
```

The default value for this setting is OFF.

The setting can be used on either a Server level or a flow level. To use the setting on the *server* level, use one of the following two methods:

To apply the setting on a *flow* level, use the following command in a WebFOCUS procedure:

```
ENGINE DB2 CDC_UPDATE_BEFORE_IMAGE ON
```

Attach the procedure on a Process flow in front of a Data flow.

Journaling for an iSeries (IBM i) library can happen in one of three ways:

1. The library is initially created by Db2 with the CREATE COLLECTION collection-name or CREATE SCHEMA schema-name command, which creates the library itself, catalogs resources with the library and standard journal resources within the library, and turns on the journaling. The CREATE SCHEMA and CREATE COLLECTION commands are synonyms for each other, and there are no differences in their actual functions.
2. Add journal resources to an existing not journaling library and turn them on, effectively turning it into a Journal Library. This option also allows for non-standard names to be used in the journaling.
3. Add a QDFTJRN Data Area to an existing not journaling library. This effectively redirects journaling to happen in an alternate library (that already has journaling enabled). Thus, the not journaling library becomes a Journal Library with journals elsewhere.

Starting from release 7.7.08M a flow or a report that uses a synonym for Table Log Records will not fail with error anymore when a sequence discrepancies happens on a journal.

Note:

- If the journal contains a very large number of records, to take an advantage of new functionality which handles the large RRN on a journal, the user needs to re-create the synonyms for the Table Log Records.

These three items are described in the following sections.

Creating a Journal Library

Start the System i Main Menu screen with the user ID that will be used to run the CDC flows.

To start Structured Query Language interface, type:

```
STRSQL
```

To create a collection, type:

```
CREATE COLLECTION LIBRARY1
```

To create the Db2 table TABLE1 in the library LIBRARY1, type:

```
CREATE TABLE LIBRARY1/TABLE1(F1 INT, F2 CHAR (10 ))
```

To insert a row into TABLE1, type:

```
INSERT INTO TABLE1 VALUES (1,'Rec 1 t1')
```

To exit the Structured Query Language interface, press the functional key [PF3].

Turning a Library Into a Journaling Library

Note: Minimized journal entries are not supported by Data Migrator. Therefore, the MINENTDTA parameter in the command CRTJRN (or CHGJRN) should not be specified. If it must be specified, this parameter should use an attribute of *NONE.

To turn a library into a journaling library, use the following commands:

To create a library called LIBRARY2, type:

```
CRTLIB LIBRARY2
```

To create a journal receiver called QSQJRN0001 for the library LIBRARY2, type:

```
CRTJRNRCV JRNRCV(LIBRARY2/QSQJRN0001)
```

To create a journal QSQJRN in the library LIBRARY2, with journal receiver QSQJRN0001, type:

```
CRTJRN JRN(LIBRARY2/QSQJRN) JRNRCV(LIBRARY2/QSQJRN0001)
```

To start Commitment Control with a lock level for ALL with default journal QSQJRN in library LIBRARY2, type:

```
STRCMTCTL LCKLVL(*ALL) DFTJRN(LIBRARY2/QSQJRN)
```

To change current library to LIBRARY2, type:

```
CHGCURLIB LIBRARY2
```

To start Structured Query Language interface, type:

```
STRSQL
```

To create the Db2 table TABLE2 in the library LIBRARY2, type:

```
CREATE TABLE LIBRARY2/TABLE2(F1 INT, F2 CHAR (10 ))
```

To insert a row into TABLE2, type:

```
INSERT INTO TABLE2 VALUES (1, 'Rec 1 t2')
```

To exit the Structured Query Language interface, press the Functional key [PF3].

To ensure journaling (to turn it ON) by adding a not journaling, pre-existing Db2 table, TABLE2, to an existing Journal Library, type:

```
STRJRNPF FILE(LIBRARY2/TABLE2) JRN(LIBRARY2/QSQJRN)
```

As a result of this statement, the following message appears: “Object of type *FILE already being journaled”.

Note: Journaling can be turned on and off for individual files with the STRJRNPF and ENDJRNPF commands.

At this point, the table TABLE2 is *log enabled*, which means that journaling works for this table. You can proceed to Data Migrator, creating the Table Log Records synonym for this table and CDC flows.

Enabling Journaling on a Not Journaling Library

There might be situations when you need to keep a table or tables in one library, and the journaling in a different library. In order to get these results, the not journaling library should get connected to a Journal library.

The following names will be used for the objects in the examples below:

- LIBRARY1 is the Journal Library that was created in Creating a Journal Library, above.
- QDFTJRN is a data area, which must exist in the LIBRARY3.

To create a library called LIBRARY3, type:

```
CRTLIB LIBRARY3
```

To Create a QDFTJRN data area in the not journaling library and have LIBRARY3 point to the Journal Library LIBRARY1, type:

```
CRTDTAARA DTAARA(LIBRARY3/QDFTJRN) TYPE(*CHAR) LEN(25)
 VALUE('LIBRARY1  QSQJRN    *FILE')
```

As a result of this, the following statement message appears: “Data area QDFTJRN created in library LIBRARY3”.

Note: VALUE() is a fixed position string with schema starting in position 1, journal name in position 11 and *FILE in position 21. In other words, VALUE('SSSSSSSSSSJJJJJJJJJ*FILE') with blank padding for names less than 10.

To change the current library to the library LIBRARY2 and display the properties of that library, type:

```
CHGCURLIB LIBRARY3
```

To start the Structured Query Language interface, type:

```
STRSQL
```

To create the Db2 table TABLE1 in the library LIBRARY2, type:

```
CREATE TABLE LIBRARY3/TABLE3(F1 INT, F2 CHAR (10 ))
```

To insert a row into TABLE1, type:

```
INSERT INTO TABLE3 VALUES (1, 'Rec 1 t3')
```

To exit the Structured Query Language interface, press the Functional key [PF3].

List of Optional Commands

Change the current library to the schema you have created with following command:

```
CHGCURLIB LIBRARY1
```

To change user profile, type:

```
CHGPRF CURLIB(LIBRARY1)
```

All tables that are created within this COLLECTION are being journaled. In other words, all the tables in this library have journals and Change Data Capture enabled for them.

You can confirm the current schema by issuing the command:

```
DSPLIBL
```

The result of this command should resemble the following image:

```
Opt  Library     Type      Device      Text
 _   LIBRARY1    CUR                   COLLECTION - created by SQL
```

To review the current library, type:

```
DSPLIBL
```

Scroll down the pages to view the journaling information.

```
                      Display Object Description - Full
                                                                Library 1 of 1
Object . . . . . :   TABLE3          Attribute . . . . . :   PF
  Library . . . . . :     LIBRARY3   Owner . . . . . . . :   EDAMSH
Library ASP device . :   *SYSBAS     Library ASP group . :   *SYSBAS
Type . . . . . :   *FILE             Primary group . . . :   *NONE

Journaling information:
  Currently journaled . . . . . :   YES
  Current or last journal . . . . . :   QSQJRN
    Library . . . . . :     LIBRARY1
  Journal images . . . . . :   *BOTH
  Omit journal entry . . . . . :   *OPNCLO
  Journaling last started date/time . :   03/17/14  17:19:29
  Starting receiver for apply . . . . :
    Library . . . . . :
    Library ASP device . . . . . :
    Library ASP group . . . . . :
```

Please observe the attributes of the table TABLE3, and notice that it is located in the library LIBRARY3, yet its current journal is QSQJRN which belongs to the library LIBRARY1.

Configuring CDC in IBM Db2 Universal Database on z/OS

Note: CDC support for UDB and Db2 requires the CLI interface.

The configuration file DSNAOINI (*.ini file) should contain the proper value for the MVSATTACHTYPE attribute. This configuration file should contain either:

```
MVSATTACHTYPE=CAF
```

or

```
MVSATTACHTYPE=RRSAF
MULTICONTEXT=0
```

Review the user ID that is specified in the Data Migrator connection for the Db2 adapter. This user ID needs to have SYSADM authority and an authority for monitoring Db2 logs. The latter implies that the system table SYSIBM.SYSUSERAUTH should contain the values for the MON1AUTH and MON2AUTH fields set to Y.

To ensure that data changes are written to the log, enabling Change Data Capture for a table, enter the following Db2 commands, where *tablename* is the name of the table to be log enabled.

```
ALTER TABLE tablename DATA CAPTURE CHANGES
COMMIT WORK;
GRANT ALL ON tablename TO PUBLIC;
COMMIT WORK;
```

To verify that the tables have the logging enabled, issue the following Db2 select statement:

```
SELECT CREATOR, NAME, TYPE
FROM SYSIBM.SYSTABLES WHERE NAME LIKE '%tablename%' AND TYPE IN( 'T','P');
```

Configuring CDC in MariaDB®

MariaDB supports Change Data Capture.

Preparing the Environment for CDC with MariaDB

In order to use CDC with MariaDB, you must install Java, Connector/J, and the MariaDB client utilities. All of these components must be installed on the same system where the Reporting Server is installed. For complete documentation on configuring data adapters for MariaDB, see the *TIBCO WebFOCUS® Adapter Administration* manual.

Java

To use CDC with the MariaDB adapter, you must have Java installed. The system Environment Variables JAVA_HOME and JDK_HOME must be specified and the location of Java must be added to the PATH Environment Variable.

For example, if your Server is on Linux, add the following lines to your profile:

```
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0_91
export JDK_HOME=/usr/lib/jvm/jdk-1.8.0_91
export PATH=$JAVA_HOME /bin:$PATH
```

Connector/J

Connector/J is an interface that provides MariaDB support for Java client programs that use JDBC connections. Connector/J is a JAR file. MariaDB Connector/J .jar files are available at *https://downloads.mariadb.org/connector-java/*.

After you download the .jar file, the location of the Connector/J must be specified in either the Environment Variable CLASSPATH on the same system where Data Migrator is installed, or in a IBI_CLASSPATH variable of a Reporting Server. To specify the location of the Connector/J in the IBI_CLASSPATH variable of a Reporting Server, perform the following steps:

1. In the DMC, expand *Workspace* and expand the *Java Services* folder.
2. Right-click *DEFAULT* and select *Properties.*

   The Java Services Configuration pane opens.
3. Expand the *Class path* section.
4. In the IBI_CLASSPATH field, specify the full path file name to the MariaDB Connector J JAR file.
5. Click *Save and Restart Java Services*.

Utilities for Processing Binary Log Files

While the MariaDB database does not have to be installed on the same system where the Reporting Server is installed, the MariaDB client utilities for processing binary log files, *mysql* and *mysqlbinlog*, must be placed into a directory on the same system where Data Migrator is installed.

The version of these MariaDB client utilities must be the same as the version of the MariaDB database. The Environmental Variable PATH should list the path to the directory where the MariaDB client utilities are placed.

The maximum size of the binary log files is set in the max_binlog_size system variable, and has a maximum possible value of 1 GB. The Binary Log is flushed when its size reaches its maximum, 1 GB. As a general rule, avoid using the LONGTEXT data type in a table that is used for CDC operations as this data type has large storage requirements.

Configuring MariaDB

MariaDB enables logging at the database level by storing the history of database changes in the Binary Log Files. These Binary Log Files are used by Data Migrator for CDC operations and must use row-based logging. For example:

```
--binlog-format=ROW
```

Although it is possible to change the logging format at runtime, the best practice is to set it at the database server startup.

A user ID assigned on a connection to a MariaDB database must have either a SUPER privilege or three global privileges: REPLICATION CLIENT, REPLICATION SLAVE, and SELECT. This task is typically performed by a database administrator. One method to assign such privileges is to issue the SQL statements shown in the following examples.

To create *user1* on any host:

```
CREATE USER 'user1'@'%' IDENTIFIED BY ‘password1’;
```

To grant *user1* all privileges:

```
GRANT ALL PRIVILEGES ON *.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

To enable *user1* for reading the binary logs from any subordinate host:

```
GRANT REPLICATION CLIENT ON *.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

To enable *user1* for reading the binary logs from any master host:

```
GRANT REPLICATION SLAVE ON *.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

To enable *user1* for selecting all objects in a database named *dbname1*:

```
GRANT SELECT ON dbname1.* TO 'user1'@'%' IDENTIFIED BY 'password1';
```

Properties of a Table Log Record Synonym

The Table Log Records synonym is the synonym for a log table. These first three columns in this synonym are added automatically and used for Data Migrator Change Data Capture processing. The columns have the following functions:

- **CDC_OPER.** is the operation type (Insert, Update, or Delete)
- **CDC_TID.** is the Transaction ID
- **CDC_TIMES.** is the time stamp
- **CDC_RRN.** Applies to iSeries (IBM i) only. Is the RRN (Relative Record Number) that is assigned by DB2 to a record when a new record is added to a table.

  The internal representation of RRN in iSeries (IBM i) is an unsigned long. In a synonym, this parameter is described as a character format with length 20 characters (USAGE=A20, ACTUAL=A20).

The Configuration section of the CDC synonym consists of special parameters, which properties could be customized by a user. Each CDC parameter in the configuration section of the CDC synonym has a default value of a global variable whose name begins with &&CDC_ and is followed by the parameter name.

The following options are available:

DB Log Parameters

DATA_ORIGIN

This is set to DBMSLOG to indicate the origin of the data is a DBMS table log.

START

Indicates the starting point for reading log records.

- **CHKPT.** After the last transaction retained in checkpoint file. This is the default.
- **CUR_TRAN.** The first transaction in the DBMS log after the job started.
- **CUR_LOG.** The first available transaction in the current active DBMS log.
- **Timestamp.** Applies to iSeries (IBM i) only. The first LUW after selected point in time (YYYYMMDDHHmmss). For example. 20200825143505.

CHKPT_SAVE

Indicates if the last processed transaction should be saved in the checkpoint file.

- **YES.** Retains the last processed transaction in the checkpoint file.

- **NO.** Does not retain the last processed transaction. This option facilitates testing, because use of Sample Data or Test Transforms does not reset the checkpoint.

CHKPT_FILE

- **Blank.** This is the default. If no file name is specified, a file is created in the same application directory as the synonym with the same name as the synonym and the extension *chp*.

Indicates the location and name of the file to use to store checkpoint information in a:

- **Physical location.** The full directory and file name in the format for the operating system used. This option should be used when there are multiple flows that will access the log for the same table. Each flow should have a unique synonym with a unique name for the checkpoint file.
- **Application directory.** The file is created in the same application directory as the synonym.

Note: Checkpoint files are generated in a proprietary internal binary format. However, a viewer is available that will display a checkpoint file in readable JSON format. To view a checkpoint file, either double-click it, or right-click it and click *View*.

Listening Parameters

POLLING

Indicates the polling interval in seconds, or how often the database log is scanned. The default is 1 second.

TIMEOUT

Indicates the timeout interval in seconds. If there is no activity in this time interval, the processing will stop. A value of zero means there is no limit. The default value is 1 second.

Note that the two options above work together. For example, if POLLING is 2 and TIMEOUT is 10, the server polls the log every 2 seconds for new transactions. If there are no new transactions after 10 seconds, then polling stops.

Read Limits

MAXLUWS

Indicates the maximum number of database transactions to process before stopping the job. A value of zero (0) means all transactions. The default value is 1 transaction.

Considerations for Table Log Records Synonyms

Timestamp parameter

The Timestamp value of the CDC parameter START can be used with:

- Db2 synonyms.
- Oracle Archived Log synonyms, that is, when CDC_LOG_SOURCE is set to USE_ARCHIVED.

CUR_LOG

The CUR_LOG value of the CDC parameter START is not supported by IBM Db2 on Mainframe (IBM System z).

Additional iSeries parameters

The following three parameters, COMMIT_MODE, LOG_NAME, and LOG_LOCATION are available on iSeries platforms only, with the Db2 database.

The LOG_NAME and LOG_LOCATION parameters are used when you are required to use a remote journal, instead of the local journal.

COMMIT_MODE

Supports the transactional commitment control.

- **ON.** Normal transaction mode with a commit used. This is the default.
- **OFF.** No commit issued or DBMS uses auto-commit mode.

LOG_NAME

Indicates the name of the journal.

LOG_LOCATION

Indicates the location name.

Current library on iSeries

When using CDC on iSeries, consider changing current library.

Example:

Start the server and the Data Management Console. From a user ID that is a member of the Server Administrator role, expand the *Workspace* folder. Expand the *Configuration Files* folder and double-click *Server Profile - edasprof.prf*. Look for a line that resembles the following:

```
-SYSTEM CHGCURLIB LIBRARY1
```

Ensure that the parameter, in this example LIBRARY1, is the name of your collection. If you made a change to file, click the *Save* icon to save the changes.

Checkpoint file on z/OS

On a mainframe the checkpoint file should be created first. This can be done by creating two focexec procedures and running them simultaneously: one procedure to report against a Table Log Records synonym and the other procedure will issue an update against the underlying actual table. For more information, see *Creating a Checkpoint File on z/OS* on page 663.

Tutorial: Using Change Data Capture with Relational Databases

Change Data Capture loads data targets using only the rows that have changed since the last load. This approach is useful when dealing with large databases that would take too much time and resources to reload completely.

The source table used in this tutorial is an inventory table called *dminv* that is created by running Create Sample Procedures and Data. The source table is in Db2 or UDB.

The tutorial has the following steps:

- ❏ Create a target table and perform an initial load.
- ❏ Create a synonym for table log records.
- ❏ Create a Data Migrator direct load flow.
- ❏ Update the source table.
- ❏ Run the flow and view results.
- ❏ Schedule the flow.

Example: Creating a Target Table and Performing an Initial Load

The first step is to create a data flow that creates the target table and loads it with data from the current contents of the source table.

The user ID to run flows using CDC should have the role of Administrator on the Server.

1. Create a new data flow in an application directory that is in the server path, like baseapp.
2. Select *dminv* as a source.
3. Select all the columns from *PROD_NUM* through *COST*.
4. Add a new target, selecting a database type, entering newinv as the name, and specifying one key.

5. Save the flow as *dminv_init*.
6. Run the flow.
7. Open the *newinv* target and click *Sample Data* to confirm that the data was loaded correctly.

Note: Key columns are required for loading a table to uniquely identify each row so that Insert/Update/Delete processing can be performed.

Example: Creating a Synonym for a Log Table

A synonym for a log table allows reading from the database log records for a selected table. The synonym contains the same columns as the table itself plus three additional columns that identify changes to the table.

This synonym can be used as a source in a data flow with the special load type of Change Data Capture.

Note: A synonym for a log table is not a substitute for a synonym for the table itself. A synonym is still needed for the table to be able to read or write from the table directly.

To create a synonym for a log table, in the DMC:

1. Right-click an application directory in the navigation pane, point to *New*, and click *Synonym (Create or Update)*.

   The Select Connection dialog box opens.

2. Double-click a connection.

   The Available Objects dialog box opens.

3. From the Object Type drop-down menu, select *Table Log Records*.

   Optionally, select the Filter by owner/schema check box and for object name enter *dminv*.

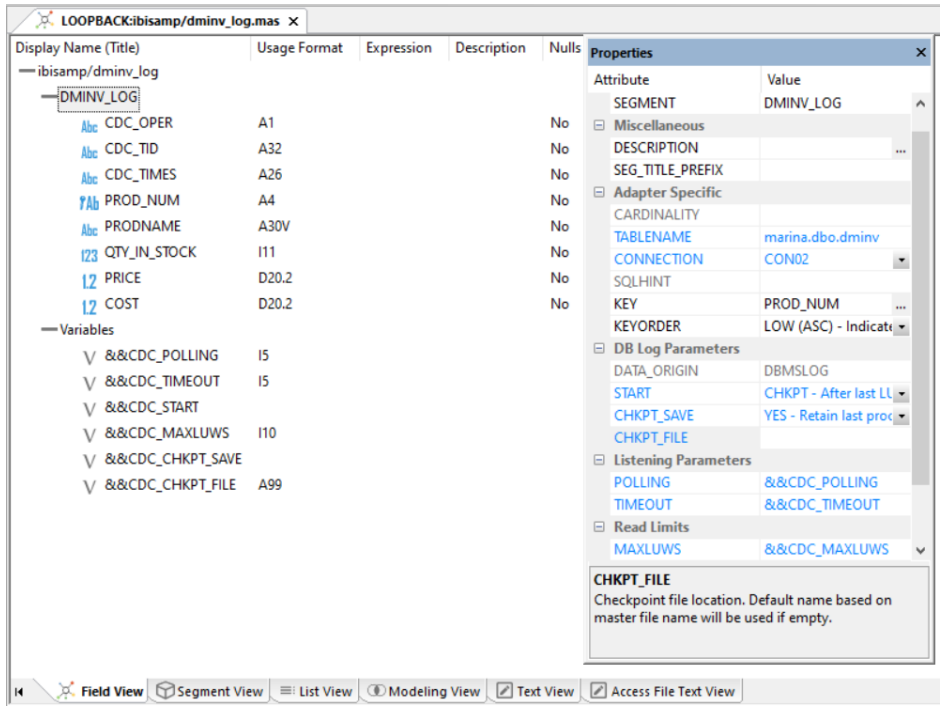
4. Select the check box in front of DMINV.

   Under the Default synonym name, click *DMINV* and change the name to DMINV_LOG.

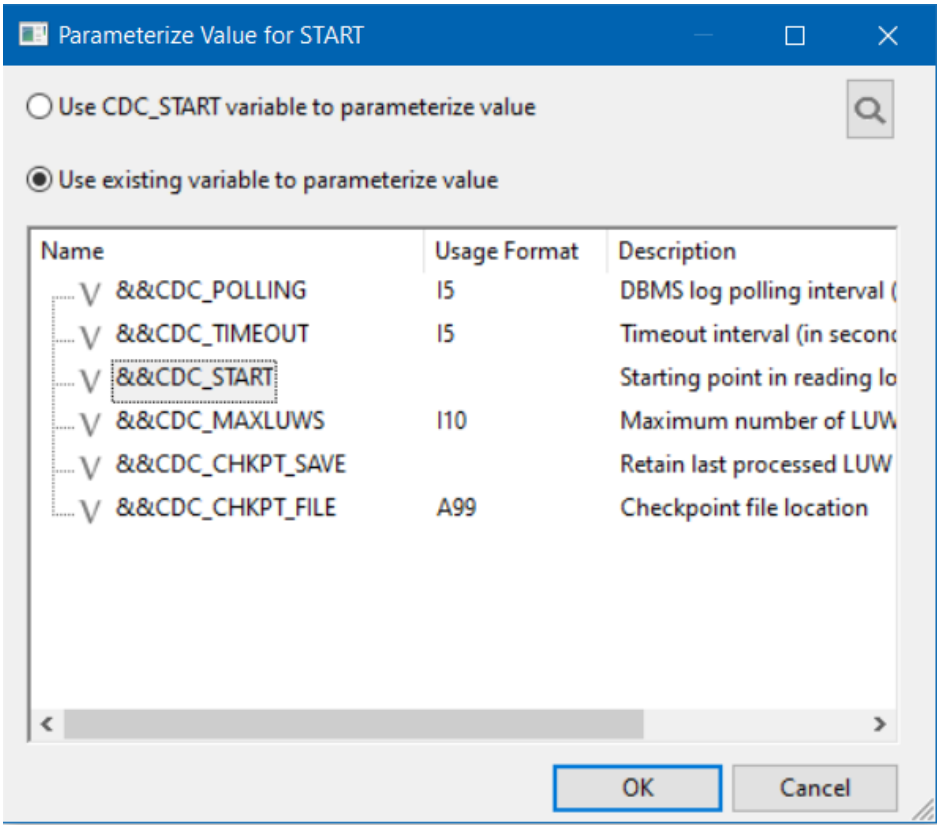
5. Click *Add*. The Create Synonym Status window opens.
6. Right-click the synonym in the navigation pane, and select Open. The synonym opens by default to the Field View. If it does not, click the tab to switch to that view. Select the segment name if it is not already highlighted.
7. In the DB Log Parameters settings, input *CHKPT* for START, *YES* for CHKPT_SAVE, and leave CHKPT_FILE blank.

Tip: Another way to change these configuration settings to the synonym is as follows. Leave the variables for the parameters values as: &&CDC_START, &&CDC_CHKPT_SAVE, and &CDC_CHKPT_FILE. Then, expand the *Variables* folder, right-click a variable, and click *Properties*. Enter the DEFAULT value for the variable.

The window now looks like this:



Tip: The attributes in a synonym that can be specified with a variable are identified in blue color, as shown in the image above. To use a variable for the CDC attribute, right-click the attribute and click *Parameterize Value* in the shortcut menu. The Parameterize Value dialog box opens, as shown in the following image.



The following options are available:

Use variable to parameterize value
Select this option if you want to use the standard CDC global variable. For example, global variable CDC_START.

Use existing variable to parameterize value
Select this option if you want to use an existing variable, which is listed and can be selected in the dialog box

Example: Creating a Checkpoint File on z/OS

In this example, we will create a checkpoint file that will be used by a CDC flow. The checkpoint file should be created on a z/OS platform in advance, prior to CDC flow runs.

1. Create a FOCUS stored procedure to report against a Table Log Records synonym. For example:

```
-SET &&CDC_POLLING=1;
-SET &&CDC_TIMEOUT=60;
-SET &&CDC_START=CUR_TRAN;
-SET &&CDC_CHKPT_SAVE=YES;
-SET &&CDC_MAXLUWS=0;
-SET &&CDC_COMMIT_MODE=OFF;
-SET &&CDC_CHKPT_FILE='baseapp/mycheck1.chp' ;
TABLE FILE LOG_MTB7
PRINT CDC_OPER FIELD2 FIELD3
END
-RUN
```

2. Create a FOCUS stored procedure to update the underlying table. For example:

```
SQL DB2
UPDATE MYETL.MYTABLE SET FIELD3 = FIELD3 +1 WHERE FIELD2 =2 ;
END
-RUN
SQL DB2
COMMIT;
END
-RUN
```

3. Run the first stored procedure - to report against a Table Log Records synonym. At the same time, run the second stored procedure – to update the underlying table. The file mycheck1.chp will be created.

Example: Creating a Data Migrator Direct Load Flow

In this step, we will create a Data Migrator direct load flow to copy data from the log table to the target table.

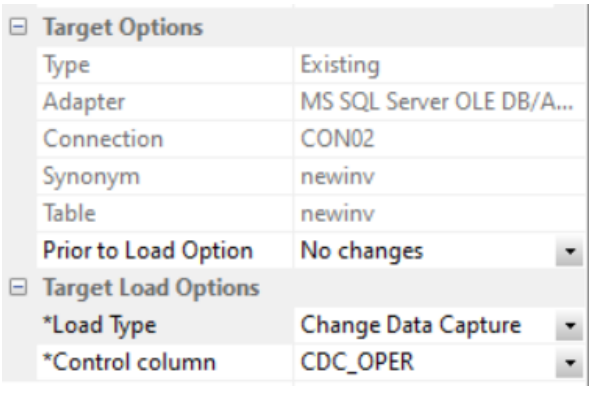
Note: Using a direct load flow with the checkpoint option ensures that all rows read from the log are written to the target table before the checkpoint file is updated. This ensures that the flow can be re-run from the original checkpoint if it cannot write to the target table (for example, if the database is not running).

While a data flow can also be used, it does not provide this assurance. The read and write operations are separate so that the checkpoint file is updated when the log is read.

1. Right-click an application directory, point to *New*, and click *Direct Load Flow*.

2. Drag *dminv_log* to the left side of the flow.
3. Drag *newinv* to the right side of the flow.
4. Right-click *newinv* and select *Properties*.

   The target properties page opens.
5. For the Load Type drop-down menu, select *Change Data Capturrre*.
6. For the Control column drop-down, select *CDC_OPER*, as shown in the following image.



   Click the *X* to close the properties page.
7. Right-click *newinv* and select *Target Transformations*.

   The Transformations window opens.
8. Click the *Automap* button to map the source to target columns.

   Click *OK* to close the window.
9. Click the *Save* button and save the flow as dminv_cdc.

Example: Updating the Source Table

To test the Change Data Capture process, make changes to the dminv source table. In this example, three rows are inserted, one row is updated, and one is deleted.

1. Start the client for your database (Db2, MySQL, Oracle, or SQL Server) and connect to the server.

   For SQL Server, enter the line:

   ```
   begin transaction;
   ```
2. Enter the following lines:

```
insert into dminv values ('2001','Compact Flash',1000,50,25) ;
insert into dminv values ('2002','Memory Stick',1000,80,40) ;
insert into dminv values ('2003','Memory Stick Pro', 1000,200,100) ;
update dminv set QTY_IN_STOCK=500 where prod_num = '2002' ;
delete from dminv where prod_num = '2001' ;
```

3. For SQL Server, enter the command:

   ```
   commit transaction;
   ```

   For MySQL, Oracle or Db2, enter the command:

   ```
   commit work;
   ```

Example: Reviewing the Log

Note that the CDC configuration saves a checkpoint of the last transaction processed. However, when you run a Sample Data report using the synonym for the log table, the checkpoint is not saved.

The following image shows a Sample Data report for the log.

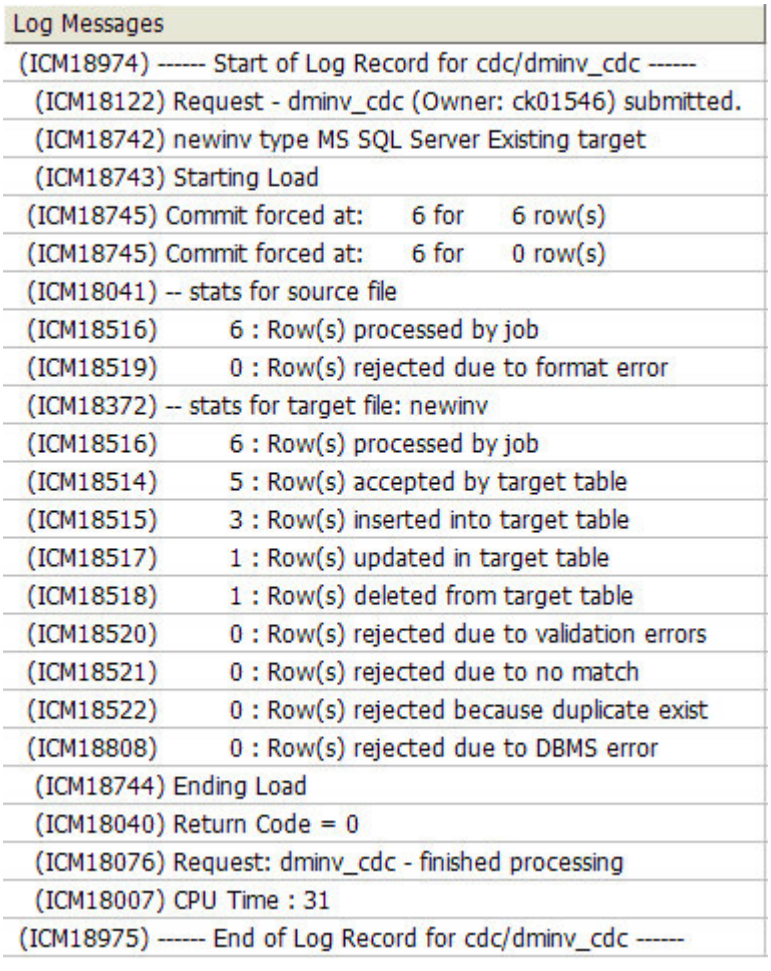


	CDC_OPER	CDC_TID	CDC_TIMES	PROD_NUM	PRODNAME	QTY_IN_STOCK	PRICE	COST
1	I	0000000045B280000000000000000000	07/02/2009 10:57:01.000000	2001	Compact Flash	1000	50.00	25.00
2	I	0000000045B280000000000000000000	07/02/2009 10:57:01.000000	2002	Memory Stick	1000	80.00	40.00
3	I	0000000045B280000000000000000000	07/02/2009 10:57:01.000000	2003	Memory Stick Pro	1000	200.00	100.00
4	U	0000000045B280000000000000000000	07/02/2009 10:57:01.000000	2002	Memory Stick	500	80.00	40.00
5	D	0000000045B280000000000000000000	07/02/2009 10:57:01.000000	2001	Compact Flash	1000	50.00	25.00
6	C	0000000045B280000000000000000000	07/02/2009 10:57:01.000000			0	.00	.00

Example: Running the CDC Flow and Viewing the Results

1. Open the CDC flow you created. Click the *Run* button and select *Submit*.

2. Click the *View Last Log* button. The following image shows the log report for the flow.

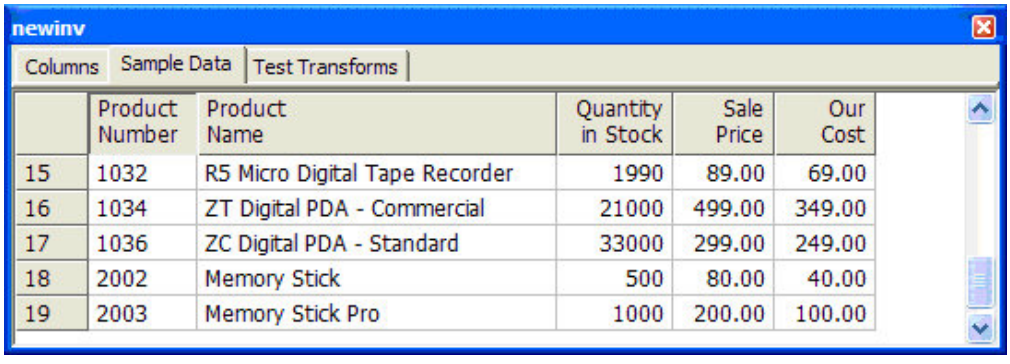


Log Messages
(ICM18974) ------ Start of Log Record for cdc/dminv_cdc ------
(ICM18122) Request - dminv_cdc (Owner: ck01546) submitted.
(ICM18742) newinv type MS SQL Server Existing target
(ICM18743) Starting Load
(ICM18745) Commit forced at: 6 for 6 row(s)
(ICM18745) Commit forced at: 6 for 0 row(s)
(ICM18041) -- stats for source file
(ICM18516) 6 : Row(s) processed by job
(ICM18519) 0 : Row(s) rejected due to format error
(ICM18372) -- stats for target file: newinv
(ICM18516) 6 : Row(s) processed by job
(ICM18514) 5 : Row(s) accepted by target table
(ICM18515) 3 : Row(s) inserted into target table
(ICM18517) 1 : Row(s) updated in target table
(ICM18518) 1 : Row(s) deleted from target table
(ICM18520) 0 : Row(s) rejected due to validation errors
(ICM18521) 0 : Row(s) rejected due to no match
(ICM18522) 0 : Row(s) rejected because duplicate exist
(ICM18808) 0 : Row(s) rejected due to DBMS error
(ICM18744) Ending Load
(ICM18040) Return Code = 0
(ICM18076) Request: dminv_cdc - finished processing
(ICM18007) CPU Time : 31
(ICM18975) ------ End of Log Record for cdc/dminv_cdc ------

Note that the statistics show the number of rows inserted, updated, and deleted.

Close the report.

3. In the open flow, dminv_cdc, right-click the *newinv* target table, click *Expand*, and select the tab *Sample Data*. The following image shows that the changes were applied correctly.



	Product Number	Product Name	Quantity in Stock	Sale Price	Our Cost
15	1032	R5 Micro Digital Tape Recorder	1990	89.00	69.00
16	1034	ZT Digital PDA - Commercial	21000	499.00	349.00
17	1036	ZC Digital PDA - Standard	33000	299.00	249.00
18	2002	Memory Stick	500	80.00	40.00
19	2003	Memory Stick Pro	1000	200.00	100.00

Note: To return the dminv table back to its original state, enter the following lines in a stored procedure, where *db* is the engine used, and run them:

```
SQL db delete from dminv where prod_num > '2001' ;
END
SQL db commit;
END
```

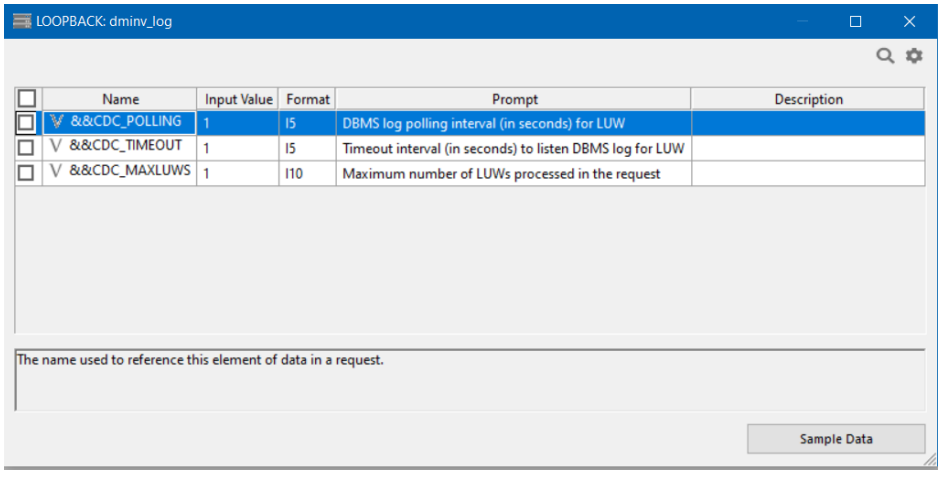
Example: Scheduling the Flow

The final step in setting up Change Data Capture processing is scheduling the flow so that it runs periodically and processes transactions.

1. Switch to the process flow view by clicking the *Process Flow* tab.
2. Right-click the *Start* object and click *Properties*.
3. Select a Schedule type, Interval type, and Interval value to run the flow.
4. Save the flow.

Reference: Sample Data

When you request sample data for a log table, a dialog box opens so that you can supply values to override the parameters in the synonym.



For example, you can sample data from a log table without creating a checkpoint by checking the check box in front of &&CDC_CHKPT_SAVE and entering the value NO.

You can also specify where to start reading the log by specifying a value for &&CDC_START. The values are the same as those specified in the synonym.

Appendix A

Log Statistics and Tables

This appendix provides information about the Data Migrator internal data store for ETL log and statistics.

In this appendix:

- Log and Statistics Tables
- TIBCO FOCUS® Logs and Statistics
- CSV Logs and Statistics
- Managing Log and Statistics
- Custom Procedures for Managing TIBCO FOCUS Logs and Statistics
- Description of the Log Files
- Description of the Statistics Files
- List of Essential ETL LOG Message Codes

Log and Statistics Tables

Data Migrator jobs can be either run immediately or submitted to the Scheduler. When a job is run, the message from the job appears in the Console Log. When a job is submitted, the messages and summary are written into the logs and statistics files by a Data Migrator service called the Scheduler. The log files contain flow run-time messages, while the statistics files contain run-time statistics for each flow. You can get access to the logs and statistics files in a read-only mode. The structure of these files is subject to change.

You can view or edit the configuration setting for the logs and statistics from the Workspace folder by right-clicking *SCHEDULER* and clicking *Properties*. The Scheduler Configuration dialog box opens.

Note: You must be logged in as a Server Administrator with workspace configuration privileges to make any configuration changes.

By default, Data Migrator creates the logs and statistics in a FOCUS file format. Logs and statistics files can be in either a FOCUS file or in a CSV (Comma Separated Values) file.

TIBCO FOCUS® Logs and Statistics

When the *sched_standalone_log* Scheduler Configuration parameter is not selected, the logs and statistics are stored in a FOCUS file. This is the default. The two FOCUS files, etllog.foc and etlstats.foc, are created by Data Migrator.

The default location for these files is in the <edaconf>/catalog/ directory path.

To create a copy of these files, you can perform the Backup operation. By default, Backup creates a backup copy of the logs and statistics inside a <edaconf>/catalog/ directory path.

To delete the etllog.foc and etlstats.foc files and to create a new pair of empty files, you can perform the Recreate operation.

Note: In release 7.6, the logs and statistics files are in FOCUS format, which is limited to 2 GB. Starting from release 7.7, these files are in XFOCUS format and the limit for this format is 32 GB.

CSV Logs and Statistics

When the *sched_standalone_log* Scheduler Configuration parameter is selected, the messages from each submitted job are written into an individual set of three standalone CSV files; one set for each submitted job.

The default location for these files is in the <edaconf>/catalog/dmlog/ directory path.

To create a copy of these files, you can to perform the Backup operation. By default, Backup creates a backup copy of the logs and statistics inside a <edaconf>/catalog/dmlogbk/ directory path.

To delete these files, you can perform the Recreate operation.

The set of the three files is: the log, the statistics, and the statistics for the Targets on a flow. The naming convention for CSV logs and statistics is:

appname_jobname_yyyymmddhhmmss_jobid_a.csv

where:

appname
Is the name of the application directory where the submitted job was located.

jobname
Is the name of the submitted job.

yyyymmddhhmmss
Is the timestamp when the job was submitted.

jobid
Is the ID assigned to the submitted job.

a
Is an indicator of the file; l - log, s - statistics, t - statistics for targets on a submitted flow.

Example: **Names for CSV Logs and Statistics**

A job name *dmdemo*, located in the application directory *ibisamp* was submitted at 11:19:01 on May 4, 2017. Three CSV files were created for this job:

- ibisamp_dmdemo_20170504111901_d497a29b_l.csv
- ibisamp_dmdemo_20170504111901_d497a29b_s.csv
- ibisamp_dmdemo_20170504111901_d497a29b_t.csv

Note: The size and the number of these files can grow according to the allowed size on a system.

Managing Log and Statistics

Note: You must be logged in as a Server Administrator with workspace configuration privileges in order to make any configuration changes, or to follow any of the following procedures.

To manage the log and statistics from the DMC, expand the*Worspace* folder, followed by the *Logs and Traces* folder. Then, right-click *Data Migrator Log and Statistics*. A shortcut menu will open with your options.

The following options are available in the shortcut menu:

View

Allows you to view the log and statistics tables.

Recreate

Creates empty new log and statistics tables.

Clean

Allows you to specify a date and time.

All records prior to the date and time entered are deleted, and the log and statistics files are rebuilt. This action frees up the space used by the deleted records, so that new records can be added. For FOCUS logs and statistics, this operation does not reduce the size of the log and statistics tables on disk, and new records can still be accepted without growing in size until the newly freed space has been freed up.

Backup

Creates a backup copy of the log and statistics tables on the disk, and a synonym that can be used to retrieve data from them. These tables are created in the <edaconf>/catalog directory that is located on the server (for example, c:\ibi\srv82\dm\catalog). A message will appear in the console showing the location.

Procedure: How to Send an Email Message When There Is an Error Writing to the ETL Log

You can configure the server so that an email message is sent to an address you specify if there is an error writing to the ETL log. For example, even though a log table is automatically maintained so that normal operations prevent it from reaching its maximum size, it is still possible for a flow to generate so many errors that it fills up the log table. This issue can be addressed by having the server send you a message when the log table is full.

1. Configure the server to connect to a SMTP mail server, as described in *How to Configure the TIBCO WebFOCUS Reporting Server for Email Notification* on page 71.
2. There are two ways to customize event routing from the DMC:

   - Right-click the *Workspace* folder and click *Event Routing*.

     or

   - Expand the *Workspace* and then the *Logs and Traces* folder.

     Right-click *Workspace Logs* and click *Customize Event Routing*.

     The Event Routing dialog box opens.

     Click .

   The Customize Event Routing dialog box opens.
3. Select the following options for each field:

   - Type - *Error*
   - Filter - *60003 ETLLOG write error*
   - EDAPRINT - *Yes*
4. In the E-mail field, enter the address where the email message will be sent.
5. In the Procedure field, enter the name of the process flow, for example, baseapp/pf_backup_recreate.
6. Click *Add*.
7. Click *Save and Restart*.

Custom Procedures for Managing TIBCO FOCUS Logs and Statistics

Note: You must be logged in as a Server Administrator with workspace configuration privileges to make any configuration changes, or to following either of the following procedures.

Procedure: How to Automatically Delete Rows From the ETL Log and Statistics Table

It may be useful for a Server to automate deleting rows from the ETL log and statistics tables. For example, you can delete all rows more than a week old on a daily basis.

1. Create a stored procedure with the following lines:

```
-SET &DDATE=AYMD(&YYMD,-7,'I8');
EX ETLDELLS DDATE=&DDATE, DTIME=00:00:00, SUSTART=Y, ETLSTART=Y
```

2. Create a new process flow.
3. Drag the procedure into the flow and connect it to the *Start* object.
4. Drag a *Schedule* icon into the workspace.
5. Double-click the *Schedule* icon and set a schedule for running the flow.
6. Save the flow.

Note: The Clean utility rebuilds the ETLLOG. The FDS service has to be stopped to prevent possible file corruption during the rebuild process. Therefore, the following considerations should be taken into account:

- ❏ No other jobs should run during an execution of the Clean utility.
- ❏ The value of the Log Destinations property of the flow should not be set to ETLLOG.

Procedure: How to Automatically Backup and Recreate the ETL Log and Statistics Tables

The server can be configured so that if there is an error writing to the ETL log, one or more of the following events will occur:

- ❏ A specified procedure is executed that will backup and recreate ETL log and statistics.
- ❏ The message is written in to the server log (edaprint.log).
- ❏ The email notifications are sent to a named address with a specified frequency.

The following procedure shows how to backup and recreate ETL log and statistics.

1. Create a new process flow and name it *pf_backup_recreate*.
2. Add the following stored procedures to the process flow in this exact order: st_backup, st_rename, and st_recreate.

The following is an explanation of the stored procedures mentioned.

st_backup

This procedure will create the *etllogbk.foc* and *etlstabk.foc* files in the \dm\catalog\ directory. These files are copies of the etllog.foc and etlstats.foc files. Example code for st_backup is:

```
SHH CRTBACKUP
END
```

st_rename

This step is optional. This procedure prevents these files from being overwritten during the next backup process. As a result of this procedure, the *etllogbk.foc* and *etlstabk.foc* files will be renamed, with a timestamp added to their suffix. For example, etllogbk_20101026_1120258.foc. Example code for st_rename is:

```
-SET &LOGBK = c:\ibi\srv82\dm\catalog\etllogbk.foc;
-SET &STABK = c:\ibi\srv82\dm\catalog\etlstabk.foc;
-SET &LOGBKREN = etllogbk || '_ ' || &YYMD || '_ ' ||
EDIT(&TOD,'99$99$99') || '.foc';
-SET &STABKREN= etlstabk || '_ ' || &YYMD || '_ ' ||
EDIT(&TOD,'99$99$99') || '.foc';
!REN &LOGBK &LOGBKREN
!REN &STABK &STABKREN
END
```

st_recreate

This procedure deletes the original etllog.foc and etlstats.foc files, and will replace them with the new initialized files. Example code for st_recreate is:

```
EX ETLCRTLS ALL, Y, N
END
```

3. Run the new process flow by clicking *Run* or *Submit one time*, or scheduling the flow. You can also use Event Routing, as described in *How to Send an Email Message When There Is an Error Writing to the ETL Log* on page 672.

Description of the Log Files

The log files contain the Data Migrator run-time message log. The number of lines that are written to this table are managed by the *sched_log_lines* and *sched_Log_commit_maxlines* Scheduler Configuration parameters.

Column	Format	Function
DATE	Alpha 20	Date and time of log entry.
USER	Alpha 32	User who created the flow.
JOBNAME	Alpha 23	Unique internal job name.
APPNAME	Alpha 256	Flow application directory.
REQ_NAME	Alpha 64	Flow name.
COUNTER	Integer 9	Line number of the message.
MESSAGE	Alpha 80	Message strings in the log.
MSGCODE	Alpha 12	Message code number.

Description of the Statistics Files

The Statistics file contains the run-time statistics for each flow.

Column	Format	Function
S_REQ_ID	Alpha 23	Name of the flow that started this flow.
S_USERID	Alpha 32	Flow author.
S_APPNAME	Alpha 64	Flow application directory.
S_REQ_NAME	Alpha 64	Flow name.
STAT_SBM_T	Alpha 20	Submit date and time for a flow.
STAT_STRT_T	Alpha 20	Start date and time of execution.
STAT_END_T	Alpha 20	End date and time of execution.
STAT_RTRV_T	Alpha 20	Date and time of retrieval for a flow.
STAT_FORMAT	Alpha 15	Number of format rejections.
REQ_RC	Alpha 16	Internal return code for the flow.

Column	Format	Function
TSCOMID	Alpha 15	TSCOMID for killing a job.
SESID	Alpha 15	SESID from EDAPRINT.
ORIG_REQ	Alpha 32	Original flow where dependency started.
ORIG_USER	Alpha 32	Original user ID running dependent flow.
DEP_CNT	Integer	Dependency number.
ETL__ERRTXT	Alpha 80	Text explanation of error messages.
FILLER	Alpha 30	Reserved for internal use. There are two columns with this field name.
USER_FILE_L	Alpha 9	Reserved for internal use.
RUN_BY_SCHED	Alpha 1	Scheduled request.
CPU_TIME	Alpha 15	Total CPU time in milliseconds.
S_TARG_NAME	Alpha 64	Target name. **Note:** This and the remaining columns apply on a per target basis.
STAT_TRANS	Alpha 15	Number of rows processed.
STAT_ACCEPTS	Alpha 15	Number of rows accepted.
STAT_NOMATCH	Alpha 15	Number of rows rejected.
STAT_DBMSERR	Alpha 15	Number of rows rejected by the database for a constraint violation, such as a Unique Index constraint, a Not Null constraint, or a user-specified constraint.
STAT_INVALID	Alpha 15	Number of rows rejected for failing validation tests.
STAT_DUPLS	Alpha 15	Number of rejected duplicate records.
STAT_INPUT	Alpha 15	Number of rows input.

Column	Format	Function
STAT_CHNGD	Alpha 15	Number of rows updated.
STAT_DELTD	Alpha 15	Number of rows deleted.

Note: You must be logged in as a Server Administrator with workspace configuration privileges in order to make any configuration changes, or to follow any of the following procedures.

In Release 7.6, the etllog file is in FOCUS format, which is limited to 2 GB. In Release 7.7, the etllog file is in XFOCUS format and the limit for this format is 32 GB.

List of Essential ETL LOG Message Codes

The following is a list of essential lines that will be imposed into the ETL LOG, even when the number of lines is limited to a positive integer.

Message Code	Log Message	Example or Explanation
ICM18007	CPU Time: %1	CPU Time : 46
ICM18031	Finished	
ICM18040	Return Code = xxx	Return Code = 0
ICM18041	– stats for source file	Statistics about processing of a source file
ICM18072	Elapsed run time hh:mm:ss	Elapsed run time 0:00:02
ICM18122	Request - %1 (Owner: %2) submitted.	Request that was saved by the owner is being submitted. The owner is saved in a flow; e.g., control variable in focexec: -*DM_USERID=user1
ICM18237	Ending bulk load procedure	Shown in the LOG when a load type on a target is 'Bulk load utility via a disk file'

Message Code	Log Message	Example or Explanation
ICM18372	– stats for target file: %1	Statistics about processing of a target file
ICM18514	%1 : %2 accepted by target table	18 : Row(s) accepted by target table
ICM18515	%1 : %2 inserted into target table	18 : Row(s) inserted into target table
ICM18516	%1 : %2 processed by job	18 : Row(s) processed by job
ICM18517	%1 : %2 updated in target table	0 : Row(s) updated in target table
ICM18518	%1 : %2 deleted from target table	0 : Row(s) deleted from target table
ICM18519	%1 : %2 rejected due to format error	0 : Row(s) rejected due to format error
ICM18520	%1 : %2 rejected due to validation errors	0 : Row(s) rejected due to validation errors
ICM18521	%1 : %2 rejected due to no match	0 : Row(s) rejected due to no match
ICM18522	%1 : %2 rejected because duplicate exist	0 : Row(s) rejected because duplicate exist

Message Code	Log Message	Example or Explanation
ICM18744	Ending Load	
ICM18750	bulkload: DBMS error code at execution	The attempt to start the bulkload failed. Check configuration. It is possible that environmental variables that point to the DBMS bulkload facilities have not been configured.
ICM18751	bulkload: Error found in the bulkload log file %1	Review the log file for DBMS messages that indicate the specific error
ICM18808	%1 : %2 rejected due to DBMS error	2 : Row(s) rejected due to DBMS error
ICM18974	Start of Log Record for %1/%2	—— Start of Log Record for baseapp/flow1 ——
ICM18975	End of Log Record for %1/%2	—— End of Log Record for baseapp/flow1 ——
ICM18995	The configured maximum number of log lines (%1) has been reached.	The message appears to inform a user that the log lines will be reduced as per the value in the sched_log_lines was set

Appendix B

Event-Based Scheduling

You can perform event-based scheduling, which executes a flow from an external application. It can be useful when you want a remote application to run a flow. For example, you might have an application that requires a flow to run. You can add a button to your application, which sends a request to the DM Server to start a flow.

Event-based scheduling can also be useful in a job that updates data. After the update, a message can be sent to the Server, instructing it to run a flow. For example, when running a batch job on USS, you can make the final step sending a message to the server telling it to execute the flow. In this instance, the flow would access the latest data.

Data Migrator supports event-based scheduling with the following:

- CMRUN is a stand-alone program that initiates flows. It is available on all supported platforms. Sample scripts (for example, shell, batch, DCL, or JCL depending on your platform) are provided for setting the environment and submitting the flow.
- A FOCUS client using specific FOCUS code can initiate a flow.

Note: CMRUN is the preferred method, and should be used whenever possible.

In this appendix:

- Starting Flows With CMRUN
- Starting Flows With CMASAP
- Starting Flows From TIBCO FOCUS
- Determining the Status of an Event-Based Flow

Starting Flows With CMRUN

The CMRUN program is a stand-alone program for initiating flows. It also allows you to specify values for user-defined variables. It is usually called from an z/OS JCL job stream, a UNIX shell script, or a Windows shortcut or command file that sets up prerequisite environment values, as shown in the examples that follow.

After a flow is run with CMRUN, a return code tells the calling program whether the flow executed successfully. For details on these return codes, see *CMRUN Return Codes* on page 690.

For example, on Windows, CMRUN is called from the Data Migrator installation directory, which is by default:

```
drive:\ibi\srv82\home\bin\cmrun.exe
```

Note: A communication file (odin.cfg) must be configured for the Server.

Syntax: How to Start Flows With CMRUN

```
CMRUN Server=server, [User=userid [, Password=password| , x=filename]],
Request=request [, Appname=appname] [, Wait=minutes] [, I=seconds]
  [, V=variable]  [, B=begin] [, E=end]  [, Multi= Y/N]
```

where:

server

Is the name of the Server. This value is required.

Note: The maximum length is 16 characters.

userid

Is a valid user ID for the server. It is required for a secured server.

password

Is the password associated with the user ID. Enclose a password in double quotes if it contains special characters.

Note: Password and filename parameters are mutually exclusive.

filename

Is a physical file name, full or relative pathed, of a file containing the server user ID password. On z/OS, the value is not the direct dataset name, but a DD name issued in the z/OS CMRUN JCL to point at the dataset name.

On other platforms, including Windows and UNIX, it is specified as a file name in the current directory, or as the relative or full path to a file name.

request

Is the name of the flow, which can be up to 32 characters long.

appname

Is the application directory where the flow is located.

minutes

Is the number of minutes the server waits for a flow to complete. The default is 0, which specifies not to wait.

seconds

Is the number of seconds the server waits before checking if the flow has completed. The default is 15 seconds.

variable

The syntax for the local variable parameter for the aforementioned operating systems:

```
variable="VAR1=value1, VAR2=value2..."
```

The syntax for the global variable parameter differs for different operating systems:

For MVS:

```
variable="&&VAR1=value1, &&VAR2=value2..."
```

For Windows:

```
Example 1:
variable=\"&&VAR1=value1, &&VAR2=value2...\"


Example 2:
variable="^&^&VAR1=value1,^&^&VAR2=value2..."
```

Note: The CALL command should not be for executing a batch file when this syntax used, as CALL does not support caret ^ escape character.

For UNIX:

```
variable=\"\&\&VAR1=value1, \&\&VAR2=value2...\"
```

For System i:

```
variable=\"\&\&VAR1=value1, \&\&VAR2=value2...\"
```

where:

`&&VAR1, &&VAR2`

Are the names of the variables.

value1, value2

Are the values assigned to the variables.

begin

Is the record where loading starts. This is not available for fixed format files.

end

Is the maximum number of records to retrieve from the server.

multi

Indicates if multiple instances of the flow can run at one time. The default is YES, which means multiple instances can be run at one time. NO means only one instance can run at a time. If the flow is running and an attempt is made to submit a second instance, it fails with an error message:

```
ICM (18997) Request app/name did not start: another instance already
```

Note: A parameter may be abbreviated as its initial letter. Parameters must be separated by a comma and space. Parameter pairs (for example, Server=server) should never have spaces before or after the equal sign. Syntax may vary for different operating systems. Check the examples for specifics.

Example: **Sample CMRUN Batch File for Windows**

Before calling the CMRUN executable on Windows, you must set up certain environment variables in order for the executable to run properly. This is typically done in batch files that in turn call the CMRUN executable. You can also set the variables at the machine level, or your DOS session, so they do not need to be directly included in any batch files that you create. Additionally, the PATH variable is often adjusted to avoid using the full path name of the CMRUN executable. The following is an example of how to set these variables, using the typical default file install and configure locations, in addition to setting the PATH variable:

```
set EDAHOME=C:\ibi\srv82\home
set EDACONF=C:\ibi\srv82\cln
set EDACS3=C:\ibi\srv82\cln\etc\odin.cfg
set PATH=C:\ibi\srv82\home\bin;%PATH%
```

The following is an example of a Windows batch file that calls CMRUN:

```
set EDAHOME=C:\ibi\srv82\home
set EDACONF=C:\ibi\srv82\cln
set EDACS3=C:\ibi\srv82\cln\etc\odin.cfg
%EDAHOME%\bin\cmrun s=chkserv, u=edachk, p=pass, r=test, w=5,
   v=\"&&v1=x,&&v2=y\"
if errorlevel 20 goto rc20
if errorlevel 16 goto rc16
if errorlevel 12 goto rc12
if errorlevel  8 goto rc8
if errorlevel  4 goto rc4
echo Request OK
goto end
:rc20
  echo (20) Request timed out
  goto end
:rc16
  echo (16) Request failed in execution
  goto end
:rc12
  echo (12) Request not found
  goto end
:rc8
  echo (8)  API Error
  goto end
:rc4
  echo (4) Invalid input parameter
:end
```

Note: To enable traces for CMRUN, pass the *-traceon* parameter as the first parameter. For example:

```
%EDAHOME%\bin\cmrun -traceon s=chkserv, u=edachk, p=pass, r=test,
    w=5, v=\"&&v1=x,&&v2=y\"
```

Example: Running a Flow Using CMRUN Without Waiting for Completion

In the following example, CMRUN is called and Data Migrator does not wait for the procedure to complete:

```
Run ( 1) Connecting to 'demo' as 'dm'
Run ( 2) DM Manager Server: Version 7, Release 7
ICM (18762) Job ID: 20171127113718_698b58d6
```

Example: Running a Flow Using CMRUN and Waiting for Completion

In the following example, CMRUN is called and Data Migrator waits for the procedure to complete. The procedure is executed successfully.

```
Run ( 1) Connecting to 'demo' as 'dm'
Run ( 2) DM Manager Server: Version 7, Release 7
(ICM18016) Request appdir/flowname submitted. Please, wait for request to
complete.
```

At completion, the following messages display:

```
(ICM18762) Job ID: 20171127113718_698b58d6
(ICM18764) Request appdir/flowname complete
```

In the following example, CMRUN is called and Data Migrator waits for the procedure to complete. The request fails.

```
Run ( 1) Connecting to 'demo' as 'dm'
Server: Version 7, Release 7
```

When the request fails, the following messages display:

```
ICM (18762) Job ID: 2002-11-13-18-37-28t3rp232_00000002
ICM (18764) Request decode failed; RC = 350
```

Example: Sample CMRUN Shell Script for HP UNIX (HPUX)

The following is an example of UNIX shell script that calls CMRUN:

```
#!/bin/ksh
export EDAHOME=/home/ibi/srv82/home
export EDACONF=/home/ibi/srv82/dm
export SHLIB_PATH=$EDAHOME/bin:/usr/lib:/lib
export EDACS3=$EDACONF/etc/odin.cfg
$EDAHOME/bin/cmrun.out s=EDASERV, r=runtest, u=edachk, password=pass,
   w=10, b=2, e=5, v=\"\&\&v1=x,\&\&v2=y\"
```

Note: To enable traces for CMRUN, pass the *-traceon* parameter as the first parameter. For example:

```
$EDAHOME/bin/cmrun.out -traceon s=EDASERV, r=runtest, u=edachk,
    password=pass, w=10, b=2, e=5,v=\"\&\&v1=x,\&\&v2=y\"
```

Example: Sample CMRUN Shell Script for SUN Solaris

The following is an example of SUN Solaris shell script that calls CMRUN:

```
#!/bin/ksh
export EDAHOME=/home/ibi/srv82/home
export EDACONF=/home/ibi/srv82/dm
export LD_LIBRARY_PATH=$EDAHOME/bin:/user/lib:/lib
export EDACS3=$EDACONF/etc/odin.cfg
$EDAHOME/bin/cmrun.out s=EDASERV, r=runtest, u=edachk, password=pass,
   w=10, b=2, e=5,v=\"\&\&v1=x,\&\&v2=y\"
```

Note: To enable traces for CMRUN, pass the *-traceon* parameter as the first parameter. For example:

```
$EDAHOME/bin/cmrun.out -traceon s=EDASERV, r=runtest, u=edachk,
    password=pass, w=10, b=2, e=5,v=\"\&\&v1=x,\&\&v2=y\"
```

Example: Sample CMRUN Shell Script for AIX

The following is an example of AIX shell script that calls CMRUN:

```
#!/bin/ksh
export EDAHOME=/home/ibi/srv82/home
export EDACONF=/home/ibi/srv82/dm
export LIB_PATH=$EDAHOME/bin:/user/lib:/lib
export EDACS3=$EDACONF/etc/odin.cfg
$EDAHOME/bin/cmrun.out s=EDASERV, r=runtest, u=edachk, password=pass,
   w=10, b=2, e=5, v=\"\&\&v1=x,\&\&v2=y\"
```

Note: To enable traces for CMRUN, pass the *-traceon* parameter as the first parameter. For example:

```
$EDAHOME/bin/cmrun.out -traceon s=EDASERV, r=runtest, u=edachk,
    password=pass, w=10, b=2, e=5,v=\"\&\&v1=x,\&\&v2=y\"
```

Example: Sample CMRUN JCL Batch File for USS Deployment

The following is an example of JCL that calls CMRUN under USS Deployment:

```
//CMRUN JOB (CMRUN),CLASS=A,MSGCLASS=Q,MSGLEVEL=(1,1),
// NOTIFY=DMUSER
//*
//* This JCL can be used to run Data Migrator jobs
//* You will need to provide the following information under
//* the EDAPARM DDname
//* U= Defaults to current userid
//* P= Password of above userid
//* R= Name of Data Migrator flow to run
//* W= Wait time
//*********************************************************************
//CMASAP EXEC PGM=CMRUN
//STEPLIB DD DISP=SHR,DSN=qualif.SRV82.HOME.LOAD
//EDAPARM DD *
S=LOOPBACK, USER=edachk, p=passwd, R=request1, APPNAME=baseapp, W=1,
 V="&&VAR1='abc',&&VAR2=123"
//EDAENV DD *
EDACONF=/home/ibi/srv82/dm
EDAHOME=/home/ibi/srv82/home
EDACS3=/home/ibi/odin.cfg
PATH=/bin
//SYSIN DD DUMMY
//STDOUT DD SYSOUT=A
//STDERR DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```

Note: To enable traces for CMRUN, pass the *-traceon* parameter with EDAPARM *ddname*. For example:

```
//EDAPARM DD  *
-traceon
S=LOOPBACK, USER=edachk,, p=passwd,, R=request1, APPNAME=baseapp, W=60
```

As a result, the cmrun.trc and tscom3.trc files will be created in your USS user directory.

Example: Sample CMRUN JCL Batch File for PDS Deployment

The following is an example of JCL that calls CMRUN under PDS Deployment:

```
//CMRUN JOB (INFO),CLASS=A,MSGCLASS=Q,MSGLEVEL=(1,1),NOTIFY=DMUSER
//* This JCL can be used to run Data Migrator jobs
//* You will need to provide the following information under
//* EDAPARM DDname
//* U=           Defaults to current userid
//* P=           Password of above userid
//* R=           Name of Data Migrator job to run
//* APPNAME=     Name of application directory for job (R=) above
//* W=           Wait time
//*
//CMASAP EXEC PGM=CMRUN,REGION=1024M
//STEPLIB DD DISP=SHR,DSN=EDAPORT.MVS.BP9999.P.HOME.LOAD
//EDACCFG DD DISP=SHR,DSN=E.MVS.BP9999.ETX.CFG
//EDAERR  DD DISP=SHR,DSN=E.MVS.BP9999.ETX.CFG
//        DD DISP=SHR,DSN=EDAPORT.MVS.BP9999.P.HOME.ERR
//CURFEX  DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(1,1,10)),
//            DCB=(RECFM=VB,LRECL=2044,BLKSIZE=27998,DSORG=PO)
//EDAPARM DD  *
S=LOOPBACK, USER=EDAPORT, p=yyy, R=xxx, APPNAME=baseapp, W=1, MULTI=Y,
Variables="v1=x,v2=y"
//EDAENV  DD  *
EDACS3=//DD:EDACCFG(ODIN)
//SYSIN     DD  DUMMY
//STDOUT    DD  SYSOUT=*
//STDERR    DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//EDAPRINT  DD  SYSOUT=*
```

Note: To enable traces for CMRUN, pass the *-traceon* parameter with EDAPARM *ddname*. For example:

```
//EDAPARM DD  *
-traceon
S=LOOPBACK, USER=edachk,, p=passwd,, R=request1, APPNAME=baseapp, W=60
```

The following example shows an alternate way to code the EDAPARM DD section:

```
// PARM=('/,s=NODENAME', ...,V=\"v1=x,v2=y\"'),
// COND=(0,NE)
```

This allows for a condition test within the JCL. However, under this style of JCL, the double quotes also require escaping with a back slash so that the double quotes literally get passed to the called procedure.

Note: The back slashes are a JCL syntax requirement.

Example: Sample CMRUN Code for IBM iSeries

The following is an example of a shell script that calls CMRUN on IBM iSeries:

```
#!/bin/qsh
export EDAHOME=home/ibi/srv82/home
export EDACONF=home/ibi/srv82/dm
export EDACS3=home/ibi/odin.cfg
liblist -a srv82
$EDAHOME/bin/cmrun.out s=EDASERV, r=request1, u=edachk,
   p=pass, APPNAME=baseapp, w=60
```

where:

`srv82`

Is the library name for edahome provided during installation

Note: To enable traces for CMRUN, pass the *-traceon* parameter as the first parameter. For example:

```
$EDAHOME/bin/cmrun.out -traceon s=EDASERV, r=request1, u=edachk,
    p=pass, APPNAME=baseapp, w=60
```

Reference: CMRUN Return Codes

When a flow is completed, CMRUN sets a return code that can be viewed in the calling environment. These codes are listed and described in the following table:

Code	Description
0	The value of the 0 code depends on the Wait parameter setting: ❏ If set to zero, the request was submitted successfully. ❏ If set to a number greater than zero, the request completed successfully.
4	Invalid input parameter.
8	Data Migrator API returned an error.
12	Request submission failed because the flow was not found, was incomplete, or the user was not authorized.
16	The request failed during execution. This return code may occur only when the Wait parameter is set to a number greater than zero.

Code	Description
20	The request timed out. This return code may occur only when the Wait parameter is set to a number greater than zero.

A return code of 8 indicates that the Data Migrator API returned an error. This error message is displayed by CMRUN. The most common errors are:

Code	Description
-7	Unable to initialize Connector. This status code indicates that the API was either unable to find or unable to read the communications configuration file. Check that the location of the correct configuration is the first one accessible using the PATH, and ensure that the contents are readable and correct
-9	Unable to connect to server. This return code generally indicates either that the server or communications resources are unavailable, or that there is an error in the configuration. Ensure that the resources indicated in the Connector configuration file are correct and available. Ensure that the server name indicated is correct and that it matches the service name on the server (for non-UNIX servers). Ensure that you can ping the host if TCP/IP is the protocol. Check to see if a connection to the server is possible from an alternate client. If this is successful, either the client configuration is incorrect or the communications resources are not available from that location. If this is not successful, ensure that the server is up and running and that it is not in an error state.
-12	Data receive failure. This status code indicates that a problem was encountered while the client was attempting to receive data from the server after an initial successful connection. Check the server status after receiving this error and check that the communications resources are intact. A -12 may occur for a specific request or for specific data retrieval.

Code	Description
-17	Invalid user ID or password. This status code indicates that a connect attempt was made to a secure server and that the user ID and/or password details supplied were not authorized for that server. A -17 may also occur if the user ID and password do not pass security exit checks set up on the server. **Note:** Some servers are case-sensitive.
-20	Expired password. This status code is issued if a user attempts to connect to the server and the password is expired.

A flow may fail during connection to the server or while being run. For details on the problems that may cause a failed flow, see *Problems, Errors, and Troubleshooting* on page 697.

Specifying an External Password

On non z/OS platforms, the X= parameter is used to point to a physical file, full or relative pathed, that contains the password for the CMRUN execution, which is protected by operating system security (read only to owner). On the z/OS platform, JCL and the X= parameter are used to point to a DD name that points to an external dataset that contains the password, which in turn can be RACF or Top Secret protected.

The file, or z/OS dataset, containing the password should contain a single line with the server password of the user ID. Passwords are case-sensitive and terminated by the first blank space or the end of the line.

Note: A password cannot contain blank spaces.

On z/OS, the dataset may be variable or fixed length , but must be of a sufficient LRECL for the actual password. Releases prior to 7706M required the z/OS dataset to be specifically LRECL=80 and RECFM=FB, but this restriction has been lifted in a way that prior password datasets also remain usable as is.

Example: Specifying an External Password With SYSIN

The following is a snip of the z/OS JCL change that is required for using an external password.

```
//CMRUN   EXEC PGM=CMRUN
//EDAPARM DD *
S=server, U=user, R=request, APPNAME=baseapp, W=10, X=//DD:PASSWD
/*
//PASSWD DD DSN=qualif.PASSWORD.DATA,DISP=SHR
```

Starting Flows With CMASAP

The CMASAP procedure enables event-oriented execution from an API program or FOCUS, and is provided with Data Migrator. It also allows you to specify values for user-defined variables.

If you need to pass parameters into an && variable, the variable must be set up with a -DEFAULT before you can use it in the execution line. For example:

```
-DEFAULT &&ODATE='&&ODATE'
EX CMASAP REQ_NAME=DMVAR, CM_ASYNC=OFF, PARMS="&&ODATE=&ODATE"
```

Syntax: How to Start a Flow With CMASAP

```
 EX CMASAP REQ_NAME=name[,APPDIR=appdir][,STOPAT=stopat][,STARTAT=startat]
 [,CM_SETERR={ON|OFF}] [,CM_ASYNC={ON|OFF}]
 [,PARMS="&&|VAR1=value1, &&|VAR2=value2..."]
[,CM_MULTI={ON|OFF}]
```

where:

appdir

Is the name of the application directory, which can be up to 64 characters long.

name

Is the name of the flow, up to 64 characters long.

startat

Is the record to start loading at. This is not available for fixed format files.

stopat

Is the maximum number of records (rows) to retrieve from the Server. RECORDLIMIT and READLIMIT are passed to the Server.

`CM_SETERR`

Determines the types of messages returned to the calling program. Valid values are:

`ON`

CMASAP sends a confirmation message and administrative messages that are produced as CMASAP prepares a flow for execution. This is the default.

`OFF`

CMASAP sends only the confirmation or failure message.

When putting an event-oriented program into production, set CM_SETERR to OFF, so that the calling program only needs to check one message when the server processes the flow.

`CM_ASYNC`

Determines if the server waits for the flow to complete. Valid values are:

`ON`

CMASAP relinquishes control to the server and terminates when the request is submitted. This is the default.

`OFF`

CMASAP waits for the flow to complete. This setting is useful for prototyping or for anytime you want to be held in a wait state until the entire flow has finished.

`&&VAR1, &&VAR2`

Are the names of the variables.

`value1, value2`

Are the values assigned to the variables.

`CM_MULTI`

Indicates if multiple instances of the flow can run at one time. Valid values are:

`ON`

Multiple instances can be run. This is the default.

`OFF`

Only one instance can be run at a time. If a second instances is attempted, it fails with the following error:

```
ICM (18997) Request app/name did not start: another instance already
```

Note: When calling this procedure from within a WebFOCUS procedure, a single pipe symbol (|) should be placed after each pair of ampersands. For example:

```
[,PARMS="&&|VAR1=value1, &&|VAR2=value2..."]
```

Starting Flows From TIBCO FOCUS

A FOCUS or WebFOCUS client on any platform is able to start flows. Notification of errors in execution will be returned to your application as error messages.

Syntax: How to Start Flows With a TIBCO FOCUS or TIBCO WebFOCUS Client

```
EX CMASAP parameters
```

where:

parameters

Are CMASAP parameters. For details, see *How to Start a Flow With CMASAP* on page 693.

Determining the Status of an Event-Based Flow

When you schedule an event-based execution, the flow is scheduled for immediate execution. If resources are available, the flow will receive its run-time parameters from the Data Migrator internal data store and begin its extraction and load as designed.

Submit the request with CMRUN. It sets a return code to the calling environment indicating if the request was submitted or completed successfully, depending on the setting for the Wait parameter.

You can also determine the status of a flow after it has been submitted. This can be done in the following ways.

- **Run the Data Migrator ETL Log (details) report.** This allows you to view the ETL Log file, which contains detailed information concerning the status of any given request. For details on the Data Migrator ETL Log (details) report, see *Generating Reports, Viewing the Log File, and Printing Flows* on page 451.
- **Run the Statistics report and check the Server statistics.**
- **Check the Web Console Workspace.** Start the Web Console and select *Agents* from the Monitor/Tune menu.

Appendix C

Problems, Errors, and Troubleshooting

This topic assists you in resolving problems that may occur during the development of your Data Migrator application.

In this appendix:

- TIBCO Data Migrator Status Codes and TIBCO WebFOCUS Reporting Server Messages
- TIBCO WebFOCUS Reporting Server Run-Time Error Codes
- TIBCO Data Migrator Management Problem Resolution
- Including Run-Time Messages in the ETL Log
- Performing a Trace to Diagnose TIBCO WebFOCUS Reporting Server Problems

TIBCO Data Migrator Status Codes and TIBCO WebFOCUS Reporting Server Messages

Error and server messages are displayed in the Message window.

The following types of messages are displayed:

- **Connector communication errors.** These error conditions usually indicate a problem communicating with the server or network.
- **Internal Data Management Console errors.** If these errors are self-explanatory, you are given the option of choosing OK or Cancel. If not, the error may indicate problems in memory or with the environment in which the DMC is operating.
- **Messages from the Server.** The most common messages are those that are sent by the Server. These will display in the message window of the DMC. Along with descriptive text, there will be a message origin and a message code.

The message typically begins with the following:

(*XXXnnnn*)

where:

XXX

Is the message origin.

nnnn

Is the message code.

Messages of this type may come directly from the subserver during the retrieval of catalog information or rows from a source table, or from the Server as administrative messages.

Common codes of the above three types are described in the following topics. Also, see the appropriate Server documentation for your platform and the Server codes at the end of this topic.

Procedure: How to Get More Information About Messages

The Server *Error Message Report* contains a list of all message numbers with a corresponding explanation. On the Workspace page, click *Settings*, then *FOCUS Sets and Info*, then *Error Message Report*. Select the error files you want to see. You can show warnings or reserved message numbers, and you can enter text to search for. Then click *Show Report*. You can also issue a query command in the DMC.

In the DMC:

1. Right-click on a server name and click *Test Server*.
2. In the section Choose Query Type, select the radio button for Focus.
3. In the procedure, type a question mark (?) followed by the message number. Then click *Test*.

   The message text and explanation appear in the Console Log area.

TIBCO WebFOCUS Reporting Server Messages

Some common server messages are listed below:

```
(FOC054)    EDA Security error
```

There is security on the synonym, and the password has not been set correctly. This should be done in the EDASPROF or the user profile for the server.

```
(FOC14010)    -column-name- IS NOT A VALID COLUMN NAME
```

DMC issues a PREPARE for the SQL command, and the column name does not match the synonym. If you have edited the synonym at the source, you may need to create the Synonym with Automatic Passthru turned off.

```
(FOC1006)    THE DDNAME REQUESTED IS NOT ALLOCATED OR DLBL'ED : table
(FOC226)    FILEDEF OR ALLOC MISSING FOR EXTERNAL FILE: table
(FOC351)    INPUT FILE NOT ALLOWED: file
(FOC1400)    SQLCODE IS  : 1006/000003EE
(FOC1405)    SQL PREPARE ERROR
```

These messages, or some combination of them, can display when an Synonym for a source file exists but the file it describes is not allocated on the server. These errors can occur upon exiting from the Query Builder or upon an attempt to run a Test Query. They can also occur when using Metadata Manager or some other tool to create a synonym.

```
(FOC1538)    4GL procedure doesn't exist: proc
```

One of the procedures required by the Server could not be found. This may indicate that a new version of the server is needed for full compatibility with the client version being used.

```
(ICM18986) Request application/name could not be found or accessed
```

This error occurs when a flow is submitted or scheduled and it can not be found in the application directory specified. Check that it exists, that your user ID can see it, and that the name is in all lower case.

```
(ICM18708) 0 rows retrieved. Stop processing.
```

This message means that the flow execution properties *Stop if 0 rows selected* is checked and zero rows were returned from the query or loaded to the target. This could indicate an error if no data was available or it could be that there were no rows that met a WHERE condition.

```
(FOC35804)    BAD STATUS CODE (-9) RECEIVED FROM SERVER subserve
```

The Server was unable to connect to a subserver listed in the server configuration file (ODIN.CFG).

```
(FOC541)  SU. CENTRAL DATABASE MACHINE ERROR: FOCSU01
```

The FDS service is down and the Scheduler is attempting to write in the ETTLOG.

Starting from Release 7.7.07, the maximum length allowed for the Application Path increased from 64 characters to 256 characters.

An example of an application path is: baseapp/app01etl/appwarehouse/history/staging.

The length of the path should not exceed 256 characters.

To enable the increased length of the Application Path, users upgrading Data Migrator to Release 7.7.07 must perform a Recreate operation for the Log and Statistics tables.

The following errors may occur when working with the application path:

```
(FOC32505) WRONG SEGMENT LENGTH DETECTED: STATS
```

The Log and Statistics tables are inherited from the older release and were not recreated.

```
(FOC015) THE TEST VALUE IS LONGER THAN THE FIELD FORMAT LENGTH
```

The full Application Path exceeds 256 characters.

TIBCO WebFOCUS Reporting Server Log Messages for Agent Termination

Server log messages for agent termination are listed below:

```
(ICM18175) Agent terminated by operator.
```

The agent was killed by an administrator or operator.

```
(ICM18174) Agent has been forced to terminate.
```

The agent was *hard killed* or crashed.

TIBCO WebFOCUS Reporting Server Log Message for a Missed Scheduled Request

The following is an example of a server message that is written if a request with a recurring schedule misses a run:

```
01/03/2010 18:04:37 E DM: procedure "baseapp/flow02" missed a scheduled run
```

To detect the run of the missed request, the Scheduler must be active and a custom routing event must be created. There is no filter needed for this event. Based on the Event Routing configuration, the server log message is written into a server log (edaprint.log) or an email message.

Connector Communications Errors

If there is a problem connecting to a Server, a message is displayed in the message window. The following are some of the more common errors:

```
Error (12029) – A connection with the server could not be established.
```

Server is not running or wrong port number specified. This error commonly occurs if the server has not started up yet. It could also occur if the host name or port number was incorrect.

```
Error (12152) – The server returned an invalid or unrecognized response.
```

This error occurs when something other than a Reporting http server is listening on the specified port number. For example, if a port number for TCP or FTP server was specified.

```
Error (32034) – Authentication failed, password is invalid.
```

This error could occur if either the user ID or the password specified were incorrect.

```
Error (32033) -- user ID is invalid Disconnected from server
```

The user ID or password that you entered was invalid for the selected server.

```
error <NGS> failed to connect to IP address:port, 10061Connection refused
```

```
Error (32007) communication error detected
```

The selected server was unavailable; it may not be running. The connector configuration (ODIN.CFG) may be incorrect.

TIBCO WebFOCUS Reporting Server Run-Time Error Codes

The errors listed here will usually display in the ETL Log unless otherwise noted. Several of these are formal Server error codes. You can see a complete list in *TIBCO WebFOCUS Reporting Server Messages* on page 698. Other errors are related to the Data Migrator environment, operating system, or specific data source system being accessed. Additional references from these resources may be necessary.

```
FOC282 -or EDA282 RESULT OF EXPRESSION IS NOT COMPATIBLE WITH...
```

This error generally occurs on a transformation string or on a DEFINE in a source synonym. It means that the answer for a calculated field cannot be placed into the column name on the left-hand side of the equal sign. Typically, this is caused by an attempt to move an alpha field directly into a numeric field (or vice versa), or a character string into a column whose format is less than the length of the desired string. For details on this and other errors in this category, see the appropriate FOCUS documentation for your server platform.

```
FOC14010    -column-name- IS NOT A VALID COLUMN NAME
```

Data Migrator issued a prepare for the SQL command, and the column name does not match the synonym. The synonym may have changed since the flow was designed. For Server Transfer Files, another possibility is that the Synonym at the Server does not match the synonym at the source. For more information, see *Designing a Data Flow* on page 171. If you have edited the synonym at the source, you may need to create the Synonym with Automatic Passthru turned off.

```
FOC35804    BAD STATUS CODE (-12) RECEIVED FROM SERVER server
FOC35862    EDA STATUS (-12): ERROR RECEIVING DATA FROM SERVER
```

These errors indicate that something has gone wrong at the server identified in the message.

One common reason for this error is that the server is running out of disk space. Check the location of the NT or UNIX temporary directory, or z/OS temporary data set allocation.

On z/OS, another possibility is that the Master and Access data sets are full. If so, delete unnecessary Master and Access numbers, and compress the data sets.

TIBCO Data Migrator Management Problem Resolution

Situational errors sometimes occur in environments with complex configurations or multiple users. The list below is meant to provide you with error messages similar to what you may encounter, and offer you resolutions such as you may find in Data Migrator. Each message is characterized by a single symptom, followed by an explanation or references to other materials.

Unique Constraint Error... Duplicate keys not permitted... or similar error for any of the relational database targets. Data Migrator has duplicate values in its extracted answer set, and there is a unique key on the table. For new targets, a unique key value may need to be specified in the Target Setup window. For existing targets, another column may need to be added to a concatenated key.

This error may also be caused by the retrieval mechanism: the use of the DISTINCT operator in the SQL command may really be the desired result (more rows than expected are being retrieved), or the source data may be a complex hierarchy or join that is creating duplicate rows. See the usage notes for your specific data target.

Format errors caused by dirty or invalid data. Legacy data is not always consistent in format and type. Older systems may have stored standard internal packed numbers for revenue figures but handled special logic conditions by inserting characters or low-values as needed by the application program. These issues often require handling with DEFINEs in the synonym at the source. There is a complete set of subroutines to handle various bit and byte oddities in your host information. For more information, see *Designing a Data Flow* on page 171, and the appropriate FOCUS documentation for your platform. Problems of this type usually make themselves known through data exceptions and other error conditions (these would display in the ETL log). Depending on the data type combinations, they may also result in rejected records or a misinterpretation of missing data.

Confusing answer set; too many rows as the result of a Cartesian product. The rules of SQL control the join process for the source data to Data Migrator. Many-to-many joins result in a Cartesian product. This may be what the warehouse requires, but very often it is confusing because it results in additional rows being generated. The Cartesian product may also result from the extraction of columns that cross two paths of a multi-path legacy hierarchy. If this occurs, reduce the number of joins, or access only one path of a complex hierarchy. For details on the Cartesian product, see the appropriate Server documentation for your platform.

Including Run-Time Messages in the ETL Log

The Dialogue Manager command -TYPE can be used to display messages. This command can be included in a procedure. If this command is used in your procedure, the text will display in the ETL Log File along with all other status and administrative messages for a flow. The text will receive a time stamp as recorded by the previous formal administrative message that displayed in the log.

Syntax: How to Generate a Message Including a Time Stamp

Use the following syntax inside your procedure every time you want a message with a unique time ID:

```
-TYPE Messages from procedure
```

Performing a Trace to Diagnose TIBCO WebFOCUS Reporting Server Problems

When you receive a server error message, you can perform a trace to find the source of the problem.

Procedure: How to Create a TIBCO WebFOCUS Reporting Server Diagnostic Package

1. In the *Information Builders*, *WebFOCUS 82 Server* program folder, click *Start Security ON/OFF With Traces* from the *Diagnostic Functions* folder to start the server with traces.
2. Reproduce the problem you encountered.
3. In the *Information Builders*, *WebFOCUS 82 Server* program folder, click *Stop*.
4. In the *Information Builders*, *WebFOCUS 82 Server* program folder, select *Diagnostic Functions*, and then click *Save Diagnostic*. Reply to the prompts to specify a location to save diagnostics and generate a list of the application directories that contain the files used to reproduce the problem.

   A diagnostics archive is created with a default name of sdyymmdd_hhmmss.zip.
5. Upload the log file.

Procedure: How to Enable or Disable Traces on a Running TIBCO WebFOCUS Reporting Server

For reporting issues to TIBCO, the complete diagnostics package described in the previous procedure is necessary. However, for diagnosing your own problem, running a single trace may be useful.

1. Using an Administrator User ID, expand *Servers* and then your server.
2. Expand the *Workspace* folder and then expand the *Logs and Traces* folder.

3. Right-click *Traces* and select *Enable Traces* or *Disable Traces*.

Procedure: How to View Traces

1. Using an Administrator User ID, expand *Servers* and then your server.
2. Expand the *Workspace* folder and then expand the *Logs and Traces* folder.
3. Right-click *Traces* and select *View*.

   The Traces window opens in the workspace.
4. Double-click the desired trace to open it.
   The trace opens in a new tab.

To view another trace, click the *Traces* tab and select the trace you wish to view.

Appendix D

Customizing Your Environment

You can control your environment by using the SET command with parameters.

The SET command is typically included in a user profile for the server or in the global profile, EDASPROF.

In this appendix:

- TIBCO Data Migrator Parameters

TIBCO Data Migrator Parameters

The following are some of the parameters that affect Data Migrator:

`MSG={ON|OFF}`

Turns on messaging for the log.

`JOINTYPE={SORTMERGE|NESTEDLOOP}`

Affects SORTMERGE or NESTEDLOOP logic.

`JOINOPT={NEW|OLD|GNTINT}`

Affects how SQL is handed to remote servers.

`CDN={ON|OFF}`

Turns on continental decimal notation for fixed format file copies.

`MISSINGTEST={SPECIAL|OLD}`

Controls whether NULL values in a comparison affect the nullability of the result.

Syntax: How to Set Parameters in a Profile or Procedure

`SET` *`parameter = option`*

where:

`parameter`

Is the setting you wish to change.

`option`

Is a valid value for the parameter.

Example: **Setting a Parameter in a Profile or Procedure**

The following sets the CDN SET parameter to ON.

```
SET CDN = ON
```

Syntax: **How to Set Target Columns as NULLABLE**

When new target tables are created, by default, the resulting target columns inherit the NULLABLE setting of their corresponding source columns.

To set the nullability of target columns, use the following syntax:

```
SET TARGET_ATTR=ON|PROPAGATE
```

where:

`ON`

Makes all non-key target columns NULLABLE (MISSING=ON).

`PROPAGATE`

All non-key target columns inherit the nullability of their source columns. This is the default.

This setting can be used in a profile, where it will affect all new target tables.

Syntax: **How to Limit the Number of DBMS Errors That Appear in the Log Table**

When Data Migrator attempts to insert rows into a target relational database table, they may be rejected due to a unique index, not null, or other constraint violations. When rows are rejected, the error messages are written to the Log Table. This can fill up the log, and it may not be necessary to see the message for every rejected row to determine the problem.

The maximum number of errors written to the log can be restricted with the following command:

```
SET DBMSMSGLIMIT=nnnn
```

where

nnnn

Is the maximum number of errors.

Note: The processing of the job continues, although further errors are not logged.

This setting can be used in a profile, where it will affect all new target tables, or in a stored procedure that is executed before the data flow loads the target table.

Syntax: How to Set the Source for Oracle Timestamps

The DATETIME_PROCESS setting allows you to specify the source for the current date and time in SQL SELECT statements from Oracle tables using CURRENT_TIMESTAMP(0).

The syntax is:

```
SQL SET DATETIME_PROCESS=SERVER|DBMS
END
```

where:

SERVER

Obtains the current date and time from the server. This is the default.

DBMS

Obtains the current date and time information from the relational database. This is only effective for queries that can use automatic pass through and only those that use the function CURRENT_TIMESTAMP(0).

This syntax should be used even if it is included in a profile, which affects all queries run on the server.

If the query cannot use APT (for example, if the source is a fixed format file) the following message is generated:

```
(FOC14070) Query cannot go APT to evaluate CURRENT_TIMESTAMP
```

Procedure: How to Use Core Engine Settings

Selected server settings can be set from the Data Management Console Workspace Configuration.

1. In the DMC, using an Administrator user ID, right-click the *Workspace* folder, point to *Settings*, then *FOCUS Sets and Info*, and click *Core Engine Settings*.

   The Change Core Engine Settings dialog box opens.

2. Select *edasprof* from the Select profile drop-down menu to make changes for all users. You can also select a single user ID.
3. Click *Next*.
4. Enter or select new values for the rest of the applicable fields.

5. Click *Save*.

Appendix E

Source Management

You can use source control programs to manage flows, stored procedures, and synonyms, typically providing file backup, change tracking, and version control. They also allow multiple users to work on the same files without overwriting changes from others. A source control program typically manages access to the source code and documents all code changes. It also typically maintains a separate version of your file each time it is checked in.

You can configure source management with Git® or with source management systems that use SCC API. For information about configuring source management with Git, see *Configuring Source Management and Setting Up Source Control for Git* on page 709. For information about configuring source management with source management systems that use SCC API, see *Configuring Source Management and Setting Up Source Control That Uses SCC API* on page 725.

In this appendix:

- Configuring Source Management and Setting Up Source Control for Git
- Configuring Source Management and Setting Up Source Control That Uses SCC API

Configuring Source Management and Setting Up Source Control for Git

Git is a free, open source Distributed Version Control System (DVCS), that allows multiple developers to work on the same remote repository, while having a full copy of the remote repository locally. The adapter for Git in Data Migrator allows you to work on your local Git repositories and share files between the local and remote Git repositories.

Unlike other source control programs currently supported by Data Migrator, Git can be used in both the Reporting Server Web Console and in the Data Management Console. It is recommended that you consistently use only one source control program while working in an Application Directory, as this will make it easier for your development team.

Procedure: How to Configure and Use an Adapter for Git

To use Git with Data Migrator you need to install Git on the same machine where the Server is installed and configure an adapter for Git.

Note:

- On Windows, the Environmental Variable PATH should include the directory where the Git utilities are installed. For example, C:\Program Files\Git\usr\bin.
- To use Git with Data Migrator, a set of JAR files should be downloaded from Internet, saved on your system, and the path to these files should be added to the IBI_CLASSPATH parameter on your server. For more information on preparing the Git environment using JAR files, see the *TIBCO WebFOCUS® Adapter Administration* manual.

The adapter for Git allows you to create a local repository inside of your Application Directory and provides access to the remote Git repositories. To configure an adapter for Git:

1. Expand the *Adapters* folder in the navigation pane, followed by the *Available* folder.
2. Right-click *GIT* and click *Configure*.

   The Add GIT to Configuration dialog box opens.
3. Click *Configure*.

   When you configure the Adapter for Git, the <local> pre-configured connection is added by Data Migrator automatically. To create a new connection for a remote repository:

   a. Right-click *GIT* in the Adapters folder and click *Add Connection*.

      The Add Connection dialog box opens.

   b. Enter the following parameter values for the connection:

      GIT URL
      The full computer name or URL for the site where the remote repositories are located.

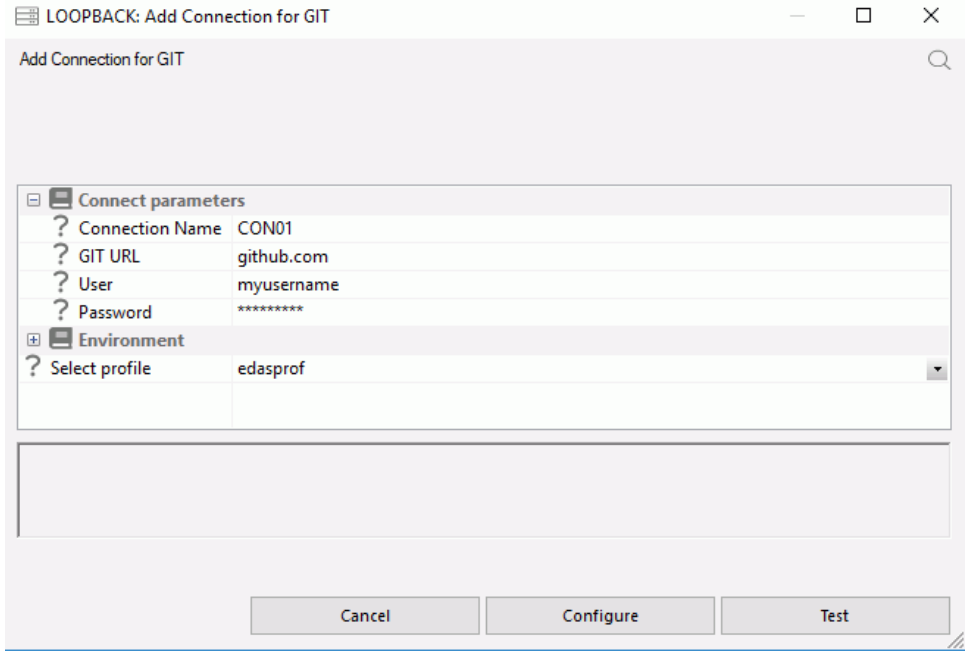
      Note: A remote repository can be located on a remote network location or on your local file system.

      User
      The user ID used to connect to the Git server.

      Password
      The password used to connect to the Git server.

      Select profile
      Select a profile from the drop-down menu to store the configured connection.

The following image shows the Add Connection dialog box with sample parameters.



Procedure: How to Set Git Options

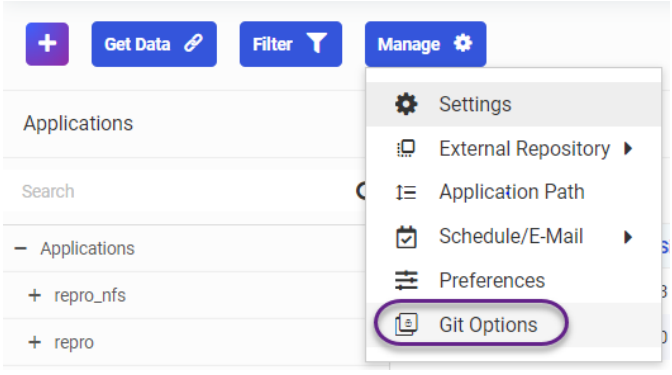
You can set the Git Options AUTHOR and EMAIL in order to associate a user with any Git commit action, or when files from the local Git repository are pushed to remote Git repositories.

You can set these options from the Web Console or the DMC.

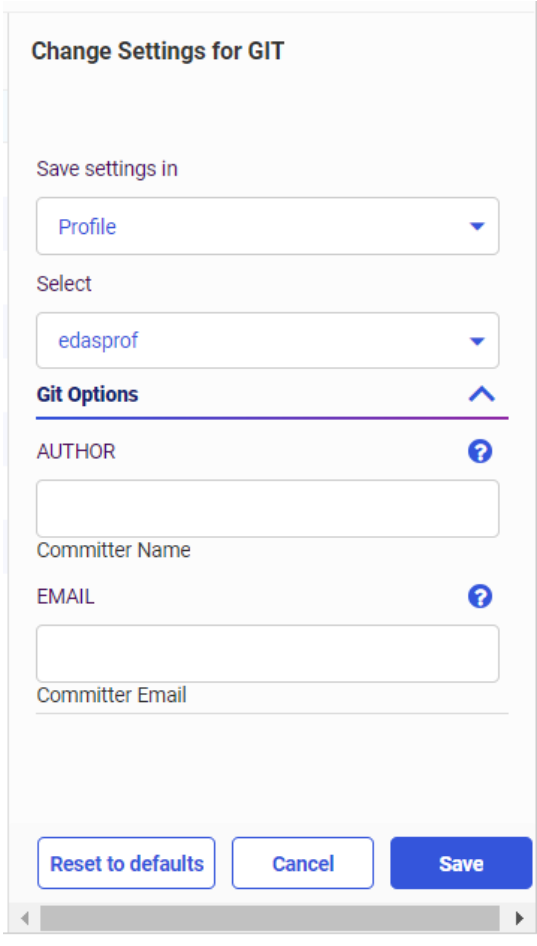
Make sure you have configured the Adapter for Git.

1. On the Web Console Applications page, click *Manage*, then *Git Options* or, in the DMC navigation pane, right-click *Application Directories*, point to *Manage*, and click *Git Options*.

The following image shows how to select *Git Options* in the Web Console.



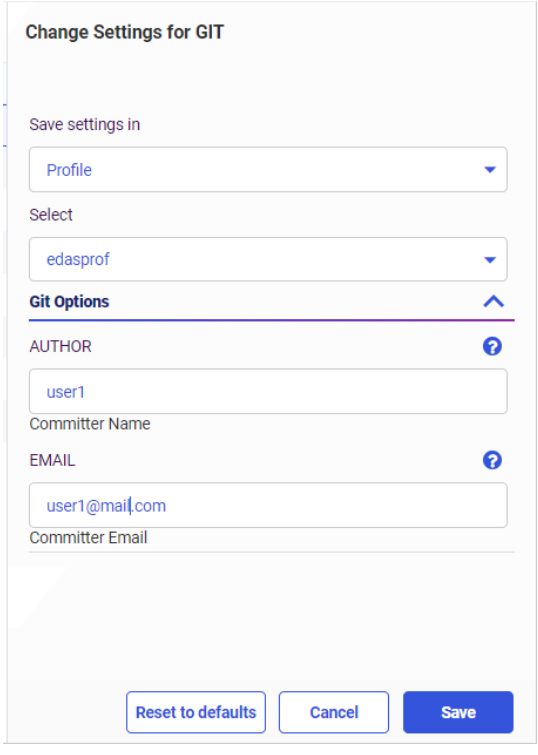
The *Change Settings for GIT* panel opens, as shown in the following image.



2. Select a profile in which to save the options.

   The default profile is the server profile, edasprof. However, you can select any profile configured in the server from the *Select* drop-down list.

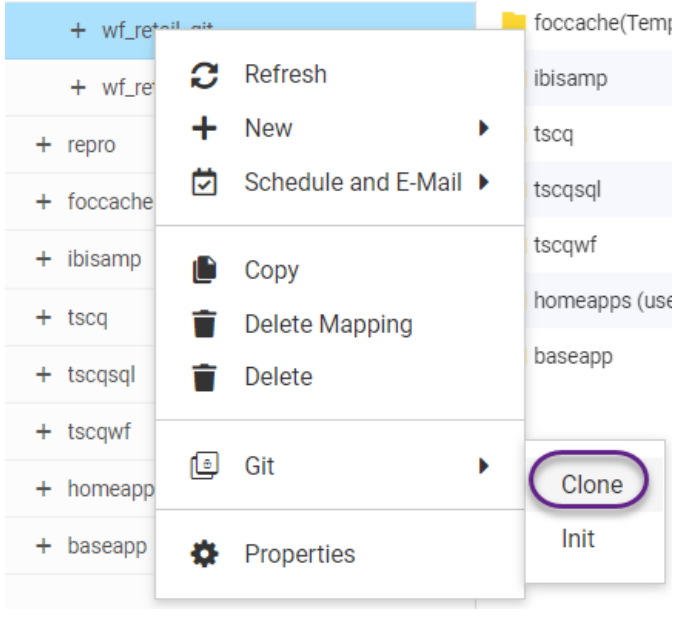
3. Enter a user name in the AUTHOR entry field and the email address of the user in the EMAIL entry field, as shown in the following image.



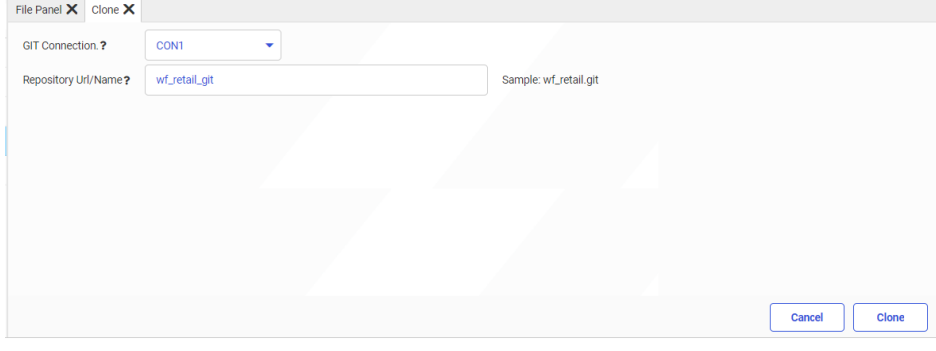
4. Click *Save*.
5. Create an empty application directory to receive the files from the central Git repository.

   This application directory is referred to as the *local repository*. The recommended best practice is to create this local repository under the *homeapps* directory of the user.

6. Right-click the local repository, point to *Git*, and click *Clone*, as shown in the following image.



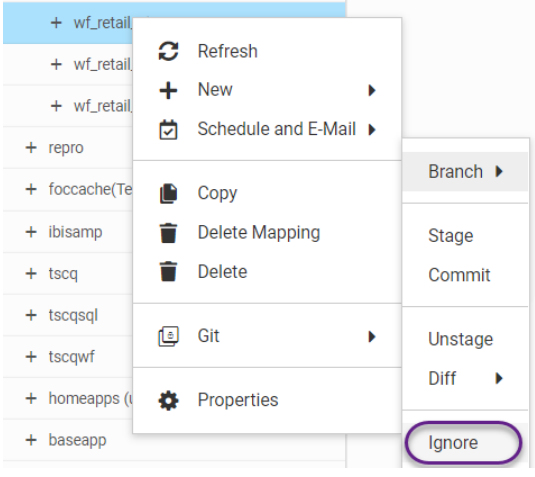
The *Clone* panel opens, as shown in the following image.



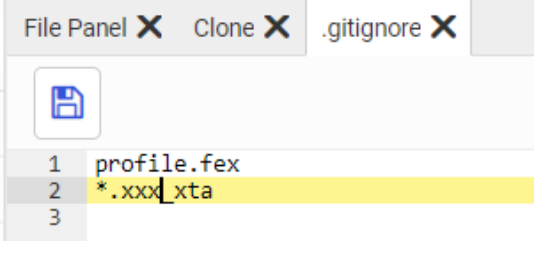
7. Select a connection from the *Git Connection* drop-down list, enter the name or URL of the Git repository in the *Repository Url/Name* entry field, and click *Clone*.

   This transfers the files from the central Git repository to the local repository on the server.

8. To exclude profile files from Git tracking, right-click the local repository, point to *Git*, and click *Ignore*, as shown in the following image.

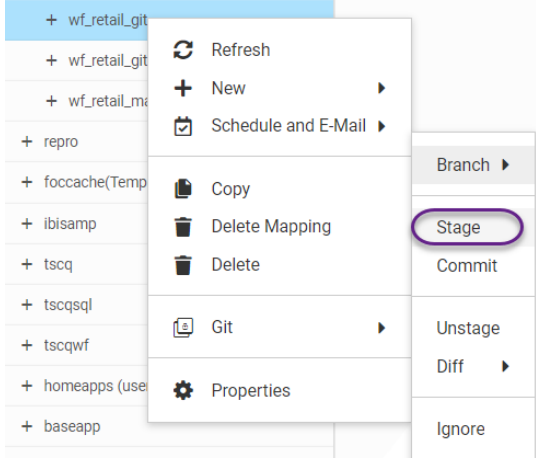


The *Git Ignore* panel opens, as shown in the following image.

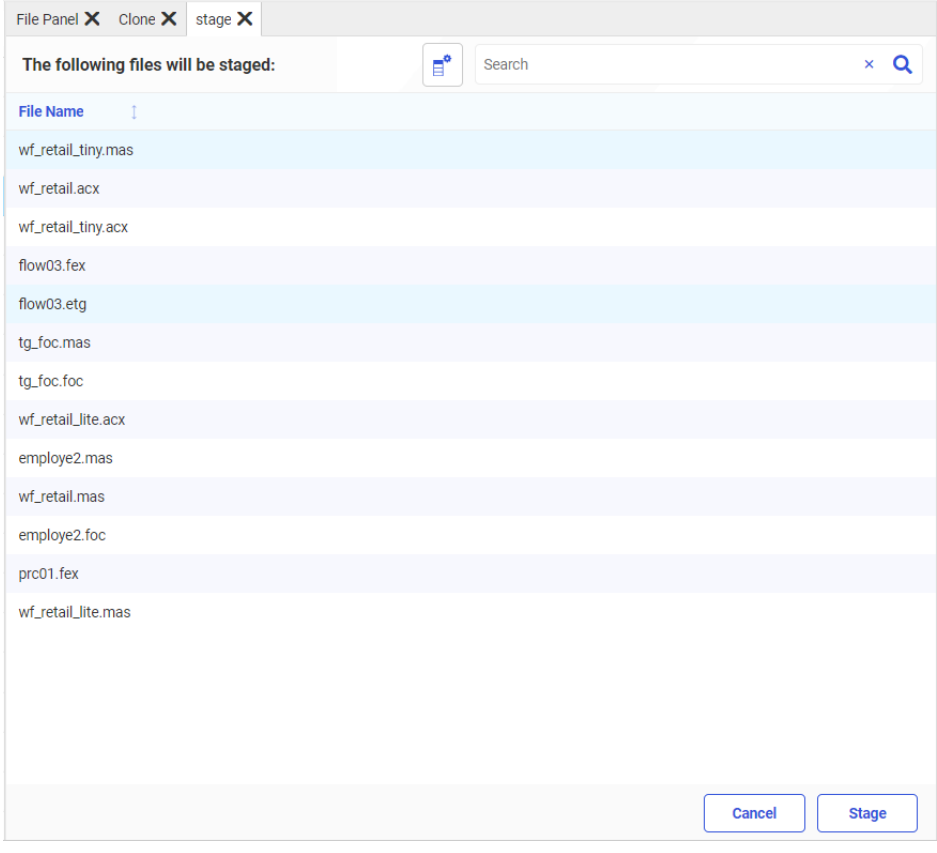


9. Enter the names or patterns that define the files to ignore, and click the *Save* icon, then close the tab using the *X* icon.
10. Make the necessary changes to any of the files in the local repository, and save the changes.

11. Right-click the local repository, point to *Git*, and click *Stage*, as shown in the following image.



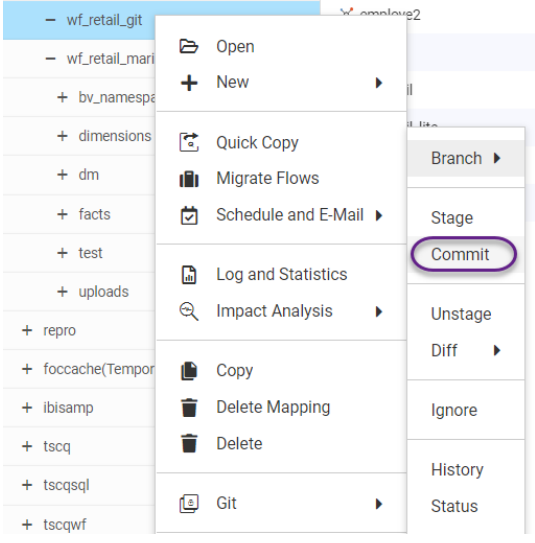
The *Git Stage* panel opens, as shown in the following image.



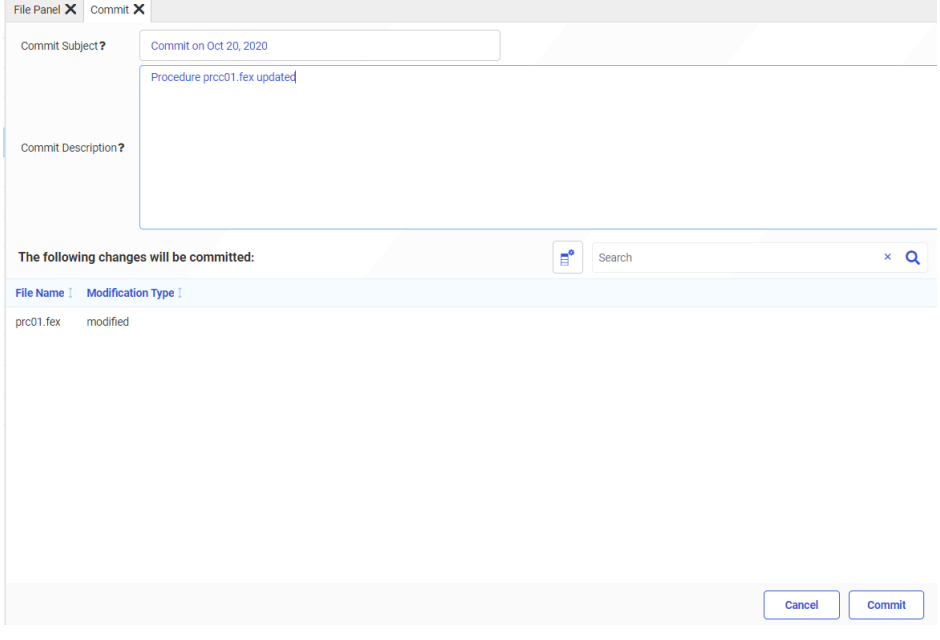
12. After you verify that the correct files are staged, click *Stage*.

    This places the files from the local repository into the Git staging area prior to committing the changes.

13. Right-click the local repository, point to *Git*, and click *Commit*, as shown in the following image.

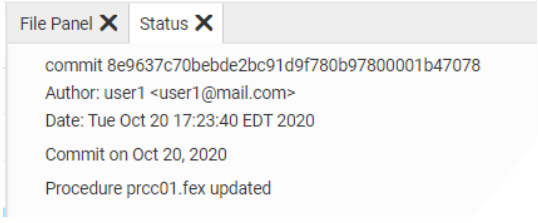


The *Git Commit* panel opens, as shown in the following image.



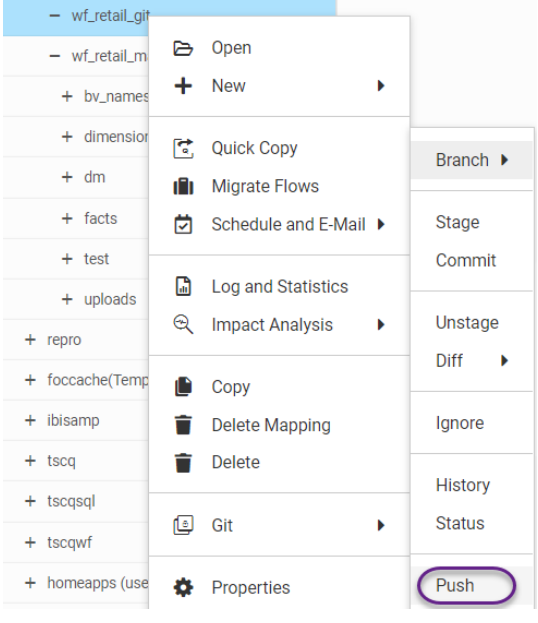
14. Enter any subject and comment text that you want to be associated with this commit, and click *Commit*.

The *Git Status* panel opens, as shown in the following image.



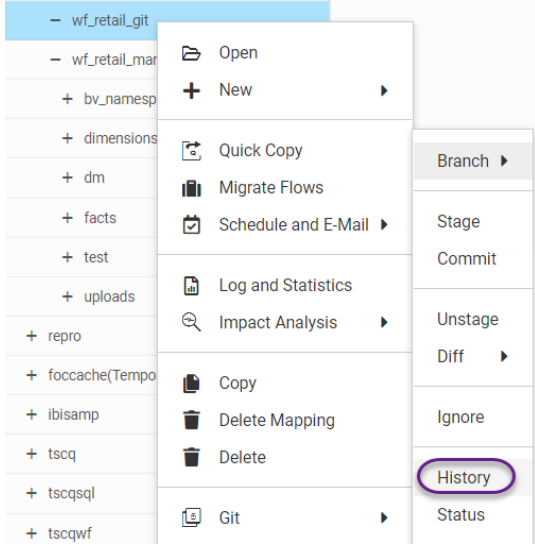
The panel shows the status of the commit, as well as the subject and comments you entered on the Git Commit panel, and the AUTHOR and EMAIL options you configured. You can close the *Status* tab using the *X* icon.

15. Right-click the local Git repository, point to *Git*, and click *Push*, as shown in the following image.

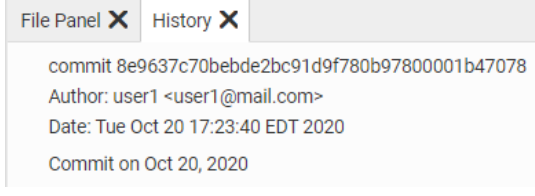


This transfers the local repository files to the central Git repository.

16. To verify that your Git Options were used, right-click the local repository, point to *Git*, and click *History*, as shown in the following image.



The history panel opens showing the information about the commit, including the AUTHOR and EMAIL Git Options you configured, as shown in the following image.



You can close the *History* tab using the *X* icon.

Procedure: How to Create a Git Repository

It is recommended that users work on their user's home directory, in *homeapps*. The *homeapps* applications are visible and available only to the connected users and server administrators. For more information on how a user creates a home directory, see *How to Create and Configure User Home Directory* on page 749.

You can start working on your local repository by performing either *Init* or *Clone* operations.

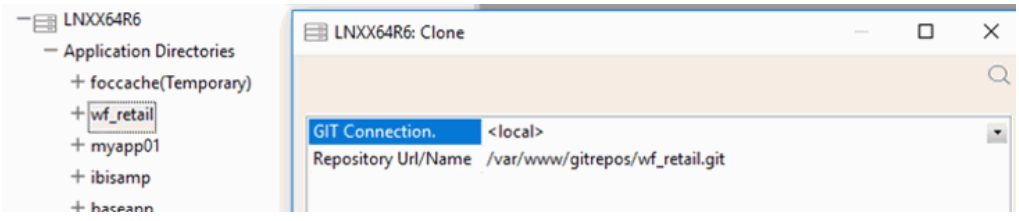
Note:

- For this procedure, your application directory must be empty.

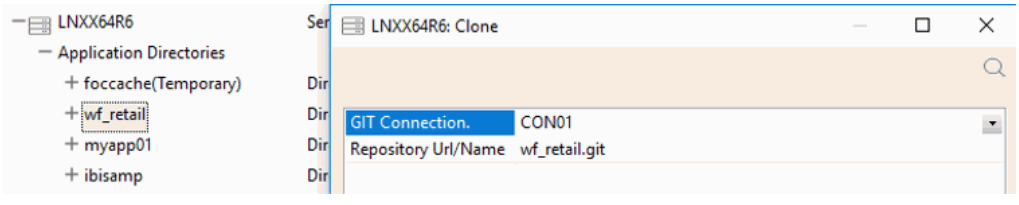
- File and application directory names on the server must be lowercase, and all file names must include an extension.

1. To create a local Git repository by cloning an existing remote repository, right-click the application directory, point to *Git*, and then click *Clone*.
2. Select the connection and enter the value for the Repository URL/Name field.

   The image below illustrates an example of the Clone operation with the default <local> connection. The repository is accessed by Git using the local protocol.

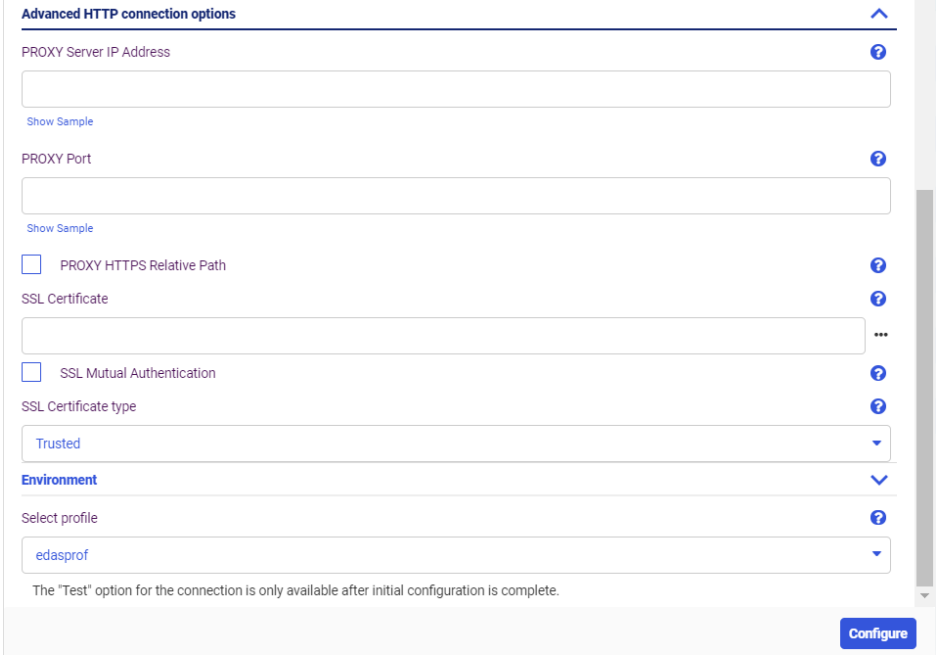


   The image below illustrates another example of Clone operation. In this example, the connection, added by a user, is set for the repository that is accessed by Git using the HTTP protocol.



Procedure: How to Configure a Connection to a Git Server That Hosts a Proxy Server With an SSL Certificate.

To connect to a Git server that hosts a proxy server with an SSL certificate, you must configure a Git connection that includes the *Advanced HTTP connection options* section of the configuration pane, in addition to the basic configuration parameters, as shown in the following image.



1. If you have not configured the Adapter for Git, configure the adapter.

   Use the instructions in *How to Configure and Use an Adapter for Git* on page 709.

2. If you already configured the Adapter for Git, and have existing connections that do not use a Proxy Server, create a new connection for this configuration.

   Configure the basic connection parameters, as described in *How to Configure and Use an Adapter for Git* on page 709.

3. Place the certificate files provided by your System Administrator in a directory on your network.

   You need to know and be responsible for whichever certificates are required and trusted. Different Certificate Authority (CA) vendors have different requirements.

4. Enter the following parameters in the Advanced HTTP connection options section.

   PROXY Server IP Address
   Is the IP address of the proxy server, or the DNS name of the proxy server, without the protocol. For example:

   ```
   myproxy.myserver.com
   ```

   PROXY Port
   Is the port on which the proxy server listens. The default value is 80.

   SSL Certificate
   Add any root and intermediate CA certificates required to validate the SSL certificate of the git repository you are connecting to. There should be one certificate per file. File names must be separated by commas, with no spaces. For example:

   ```
   /users/user1/cert1/c1.crt,/users/user1/cert1/c2.crt,/users/user1/
   cert1/c3.crt,/users/user1/cert1/c4.crt
   ```

   SSL Certificate Type
   Select *Trusted*.

5. Optionally, to test the connection, click *Test*.

   A message should be returned indicating the the connection is valid. If there are errors, correct them.

6. When the connection attributes are valid, click *Save*.

   The connection is added to the Adapter for Git.

Procedure: How to Update a File in Your Git Repository and Send Your Changes to the Remote Repository

Once your Git connection is set up, you can work in a file located in your application directory and then send (or push) the changes to the remote repository, using the following steps:

1. Edit a file in the application directory where you created a repository, click *Save*, and close the file.
2. To move your file into the staging area in your local Git repository, right-click the file, point to *Git*, and then click *Stage*.
3. To commit your changes, right-click the file, point to *Git*, and then click *Commit*.

   The Commit dialog box opens.

   Enter any desired commit comments and click *Commit*.

4. To send your changed file to the remote repository, right-click the application directory, point to *Git*, and then click *Push*.

Configuring Source Management and Setting Up Source Control That Uses SCC API

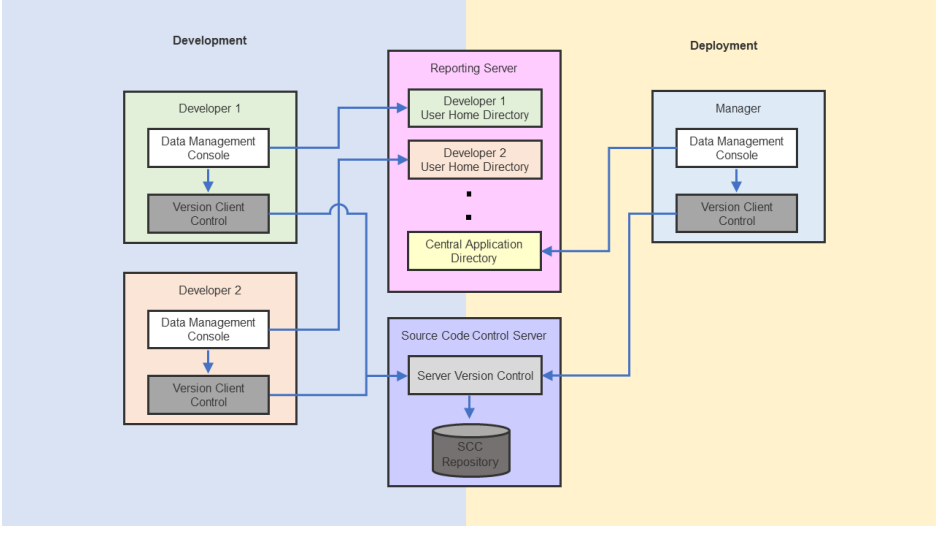
If you have a source control system that uses SCC API configured, you can use it to source manage Data Migrator procedures, synonyms, and flows.

Note: The Git source management system is configured as an adapter, not as described in this topic. For information, see *Configuring Source Management and Setting Up Source Control for Git* on page 709.

Data Migrator provides access to basic source management functions through right-click menu options in the navigation pane which only appear if you have a source control system configured.

Source control systems consist of a repository that holds a master copy of each revision. The local working copies of these files are stored on the DM server inside the user home directory. These files are the files changed by developers, before the file is committed or checked-in to the source control repository, which creates the next revision. Data Migrator can add synonyms, procedures, and flows to source control by exporting ACCESS, MASTER, FOCEXEC, and ETG files, respectively. Intermediate files are created in the user home directory of the developer. It is the role of the manager to create a user home directory for each developer that will contain the local working copies. When a developer checks out a flow, Data Migrator sends a command through SCC API plug-in or by invoking client command line utility to source control to check out the file. The source control system then puts the latest version of the file into the user home directory of the developer. All other functions, such as add, check in, undo check out, view history, and get latest version functions, are invoked the same way.

The intermediate work space of the manager is the actual central application directory. Only the manager has write privileges to central application directory. The manager can deploy the latest files from source control repository to the central application directory by invoking get latest version. The following diagram shows the model of the workflow.



To implement source control that uses SCC API, all developers and the manager must:

1. Set up source control folders in your source management system. These folders are called Projects in ChangeMan DS, Views in ClearCase, and Modules in CVS.

   Note: Source control folders work essentially like Data Migrator application directories. You should create folders in your source control system that mimic your node/application directory structure. Creating folders using your server node names will help you distinguish application directories with the same names on different servers.

2. Configure Data Migrator to work with your source control system.
3. Create associations between Data Migrator application directories and the source control folders you created.

You can then add files to your source control system, check them out or in, and see file history, versions, and differences from the Data Management Console.

For detailed information on installing and configuring your source control system, and setting up source control databases and projects/views/modules, see your source control system documentation.

Setup, Configure, and Associate TIBCO Data Migrator to Work With ChangeMan® DS

Creating a ChangeMan DS project provides a way to logically associate files with a project. Files are associated to the project by attaching them when the project is created.

Note: Source control folders, called projects in ChangeMan DS, work essentially like Data Migrator application directories. ChangeMan DS projects can be created during configuration or when you create associations with application directories.

Procedure: How to Create ChangeMan DS Projects

1. Log into ChangeMan DS.
2. From the Tools menu, click *System Configuration*.

   The Global Process Control window opens.
3. From the Edit menu, click *Users*.

   Select *New User* or *New Group* from the bottom left of the Global User Configuration window, and create and edit user and group profiles. Provide privileges to the users.
4. From the Edit menu of the Global Process Control window, click *Areas*.

   Select *New* from the bottom left of the Global Area Configuration window. Create and edit areas, such as development, production, QA, and end users. The development area usually represents a directory on the local computer where the development work takes place. Files may be checked in and out of this directory.
5. From the Edit menu of the Global Process Control window, click *Projects*.

   Select *New* from the bottom left of the Project Configuration window to create and edit projects. ChangeMan DS projects help to manage logically related files. Attach files to a project during configuration.

   Note: Currently, identical sources cannot be used for the same projects. This feature is planned for a future release.
6. From the Process Control Window of the new project that was created, click the *New Rule* icon located right under the File menu.

   Create a rule by adding the appropriate nodes and connecting the Development Area with the Production Area.

Procedure: How to Configure TIBCO Data Migrator to Work With ChangeMan DS

To use a source control system to source manage your Data Migrator flows, stored procedures, and synonyms, you must have a source control program and database configured. The source control program will then appear as an SCC Provider.

In the Data Management Console:

1. On the Home tab, in the Tools group, click *Configure Source Control*.

   Note: If you have not properly configured ChangeMan DS Client, this option will not appear in the Tools group.

   The Configure Source Control dialog box opens.

2. Select *ChangeMan DS* as the SSC Provider from the drop-down menu.

   Note: Selecting *none* removes source control from your system.

3. Optionally, click *Test* to make sure you can connect to the source control system from the DMC.
4. Click *OK*.

   The Set Source Control Provider dialog box opens.

5. Click *Yes*.

   Once a SSC Provider has been set, source control options appear on the right-click menus.

Procedure: How to Create Associations Between Application Directories and ChangeMan DS Projects

Creating a source control project provides a place to manage your Data Migrator files. Creating an association between your application directory and that project tells Data Migrator where to manage those files.

Note: Developers must associate application directories inside their user home directory, while only the manager associates to the central application directory.

In the navigation pane of the Data Management Console:

1. Right-click an application directory and select *Source Control*, and then click *Create Association with Source Control*.

   The Associate Application with Source Control Project window opens.

2. Click *Associate*.

   The ChangeMan DS Login dialog box opens. Type your User Name, Password, and MainServer for your ChangeMan DS environment.

3. Click *OK*.

   The SCC Project Path from ChangeMan DS dialog box opens.

4. Enter Local Area and Project and click *OK*.

5. Click *OK* to close the Associate Application with Source Control Project Dialog box.

   The Application Directory icon displays a lock to indicate the source control association.

At this point, you have associated your application in Data Migrator with a project in ChangeMan DS.

Setup, Configure, and Associate TIBCO Data Migrator to Work With ClearCase®

Source control folders, called views in ClearCase, work essentially like Data Migrator application directories. You should create folders in your source control system that mimic your node/application directory structure. Creating folders using your server node names will help you distinguish application directories with the same names on different servers.

Procedure: How to Create ClearCase Views

1. From the Start Menu, select *Rational Software*, *Rational ClearCase*, and then click *Create View*.

   The View Creation Wizard's Choose a Project dialog box opens.

2. Click *Next*.

   The View Creation Wizard's Choose Snapshot View or Dynamic View dialog box opens.

3. Select *Dynamic* and click *Next*.

   The View Creation Wizard's Choose Name and Drive dialog box opens.

4. Use the default view name or enter an alternate name.
5. Select a drive. In this example, the default drive, Z, is used.
6. Enter a comment for the view.
7. Click *Advanced Options*.

   The Advanced View Options dialog box opens.

8. Enter the name of your ClearCase storage share and click *OK*.
9. When the Choose Name and Drive dialog box reopens, click *Finish*.

   The Confirm dialog box opens.

10. Click *OK*.

    The ClearCase View Tool dialog box opens.

11. Click *OK*.

Once you have created a ClearCase view, you should navigate to its drive/folder and create subfolders that match the application directories that you want to source manage.

Procedure: How to Configure TIBCO Data Migrator to Work With ClearCase

To use a source control system to source manage your Data Migrator flows, stored procedures, and synonyms, you must have a source control program and database configured. The source control program will then appear as an SCC Provider.

In the Data Management Console:

1. On the Home tab, in the Tools group, click *Configure Source Control*.

   Note: If you have not properly configured ClearCase Client, this option will not appear in the Tools group.

   The Configure Source Control dialog box opens.

2. Select *ClearCase* as the SSC Provider from the drop-down menu.

   Note: Selecting *none* removes source control from your system.

3. Optionally, click *Test* to make sure you can connect to the source control system from the DMC.

4. Click *OK*.

   The Set Source Control Provider dialog box opens.

5. Click *Yes*.

   Once a SSC Provider has been set, source control options appear on the right-click menus.

Procedure: How to Create Associations Between Application Directories and ClearCase View Folders

Creating a source control folder provides a place to manage your Data Migrator files. Creating an association between your application directory and that folder tells Data Migrator where to manage those files.

Note: Developers must associate application directories inside their user home directory, while only the manager associates to the central application directory.

1. Right-click an application directory, select *Source Control*, and then click *Create Association with Source Control*.

   The Associate Application with Source Control Project dialog box opens.

2. Click *Associate*.

   The Browse for Folder dialog box opens.

3. Select the folder that you want to associate with this application directory and click *OK*.
4. Click *OK* to close the Associate Application with Source Control Project dialog box.

   The Application Directory icon displays a lock to indicate the source control association.

You are now ready to use ClearCase for source management.

Setup, Configure, and Associate TIBCO Data Migrator to Work With Concurrent Versions System (CVS)

CVS is an open source system widely used in the UNIX world. It has a Windows implementation called CVSNT. Since CVS is a command-line system, it cannot be used directly by Data Migrator (or any other application that uses the SCC API). To use CVSNT, the PushOK CVS SCC Proxy is needed to convert the SCC API calls into CVS commands. Source control folders, called modules in CVS, work essentially like Data Migrator application directories.

Note: The PushOK proxy is available from *http://www.pushok.com* as the CVS SCC Plug-in. Use the Windows version, not the x64 version, even if you are running 64-bit Windows.

Procedure: How to Create Concurrent Versions System (CVS) Modules

Before you can create CVS Modules, you need to create a repository and add users. A repository needs a root directory where both the files under version control and auxiliary files for use by CVS server are stored. After creating the repository, create users that can access (store and retrieve) files from the repository. Once this is complete, there are two ways to create modules:

1. You can add a new module through the command line. For instance, to add a new module with the name app1:

   ```
   mkdir $CVSROOT/app1 # makes an empty module.
   ```

   ```
   cvs checkout -d ./ app1 # checks it out to the current directory
   ```

2. Use a CVS client tool, such as TortoiseCVS. This tool allows you to work with files under CVS directly from Windows Explorer, thus eliminating the need to code. Other CVS client tools include CVSGui, SmartCVS, and many others.

Note: CVS modules can be also be created when you create associations with application directories. For details, see *How to Create Associations Between Application Directories and CVS Modules* on page 732.

Procedure: How to Configure TIBCO Data Migrator to Work With CVS

To use a source control system to source manage your Data Migrator flows, stored procedures, and synonyms, you must have a source control program and database configured. The source control program will then appear as an SCC Provider.

In the Data Management Console:

1. On the Home tab, in the Tools group, click *Configure Source Control*.

   Note: If you have not properly configured PushOK CVSSCC NT Proxy, this option will not appear in the Tools group.

   The Configure Source Control dialog box opens.

2. Select *PushOk CVSSCC NT* as the SSC Provider from the drop-down menu.

   Note: Selecting *none* removes source control from your system.

3. Optionally, click *Test* to make sure you can connect to the source control system from the DMC.
4. Click *OK*.

   The Set Source Control Provider dialog box opens.

5. Click *Yes*.

   Once a SSC Provider has been set, source control options appear on the right-click menus.

Procedure: How to Create Associations Between Application Directories and CVS Modules

Creating a source control folder provides a place to manage your Data Migrator files. Creating an association between your application directory and that folder tells Data Migrator where to manage those files.

Note: Developers must associate application directories inside their user home directory, while only the manager associates to the central application directory.

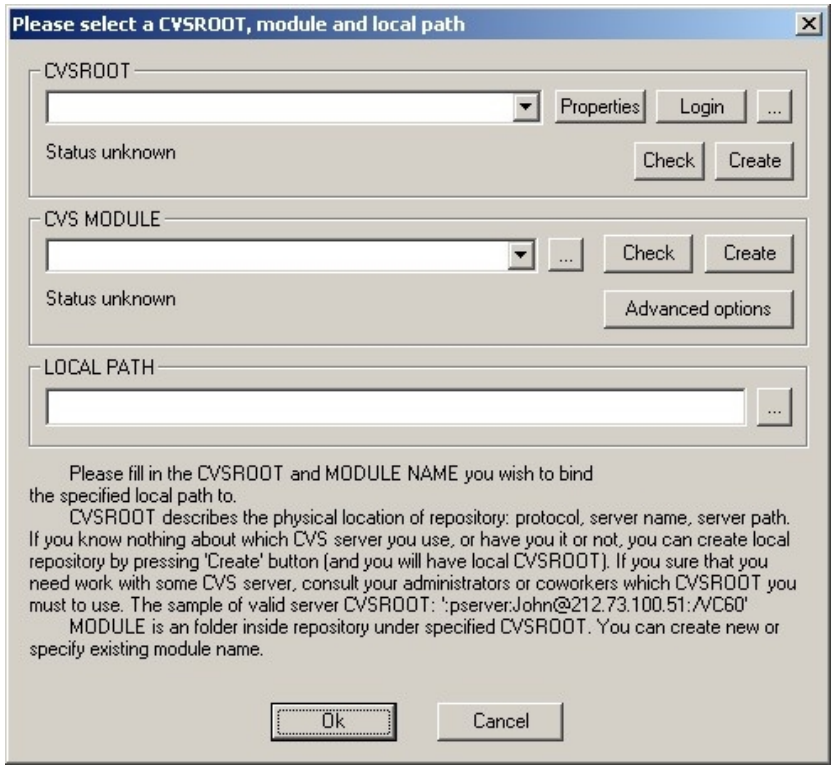
In the navigation pane of the Data Management Console:

1. Right-click an application directory, select *Source Control*, and then click *Create Association with Source Control*.

   The Associate Application with Source Control Project window opens.

2. Click *Associate*.

The Please select a CVSROOT, module and local path dialog box opens, as shown in the following image.



3. Enter the repository information in the CVSROOT field. CVSROOT describes the physical location of repository, the directory that stores the master copies of the files.

   The syntax for the CVSROOT is:

   ```
   :protocol:pcname/path
   ```

   where:

   protocol

   Is for a Windows server, the recommended protocol is sspi.

   pcname

   Is the name of the machine running CVS.

   path

   Is the location of the CVS repository.

An example of a complete syntax is:

```
:sspi:PC1/cvs
```

4. Click *Check*.

   If CVSROOT is valid, the following message should appear:

   ```
   The CVSROOT is OK.
   ```

5. Click *Properties*.

   The PushOK CVS Proxy properties window opens.

6. Click the *CVS Options* tab, deselect the *Enable file remove monitoring* check box, and click *OK*.
7. Create a new module or specify the existing module name. Module is a specific directory (tree of directories) in the main repository, CVSROOT.
8. Click *Check*.

   If Module is valid, the following message should appear:

   ```
   The MODULE is OK.
   ```

9. Set the local path to *c:\ibi\srv82\dm\etlgtemp\server*. The local path is where the source managed files temporarily reside.
10. Click *OK* to close the select a CVSROOT, module and local path dialog box.
11. Click *OK* to close the Associate Application with Source Control Project dialog box.

    The Application Directory icon displays a lock to indicate the source control association.

You are now ready to use CVS for source management.

Setup, Configure, and Associate TIBCO Data Migrator With Subversion®

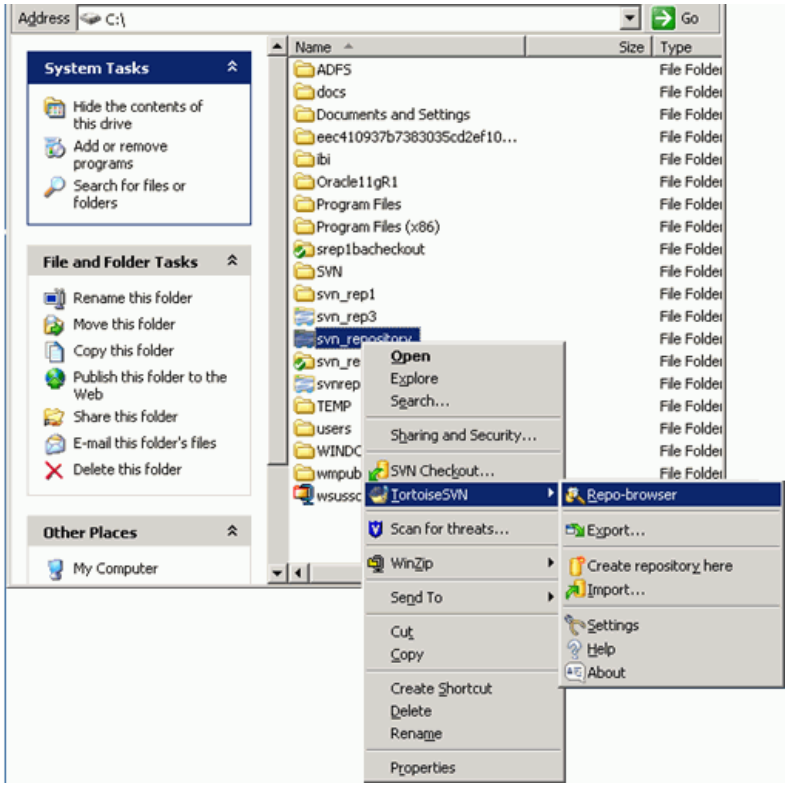
Subversion is an open source revision control system and successor to Concurrent Versions System (CVS) that can run on cross-platform operating systems. This section will discuss the Windows implementation. Subversion is a command-line system and cannot be used directly by Data Migrator. To use Subversion, PushOk SVN SCC Proxy is needed to convert the SCC API calls into Subversion commands. It is recommended that the source control folders, also known as modules, are structured the same as the Data Migrator application directories.

Note: The PushOK proxy is available from *http://www.pushok.com* as the SVN SCC Plug-in. Use the Windows version, not the x64 version, even if you running 64-bit Windows.

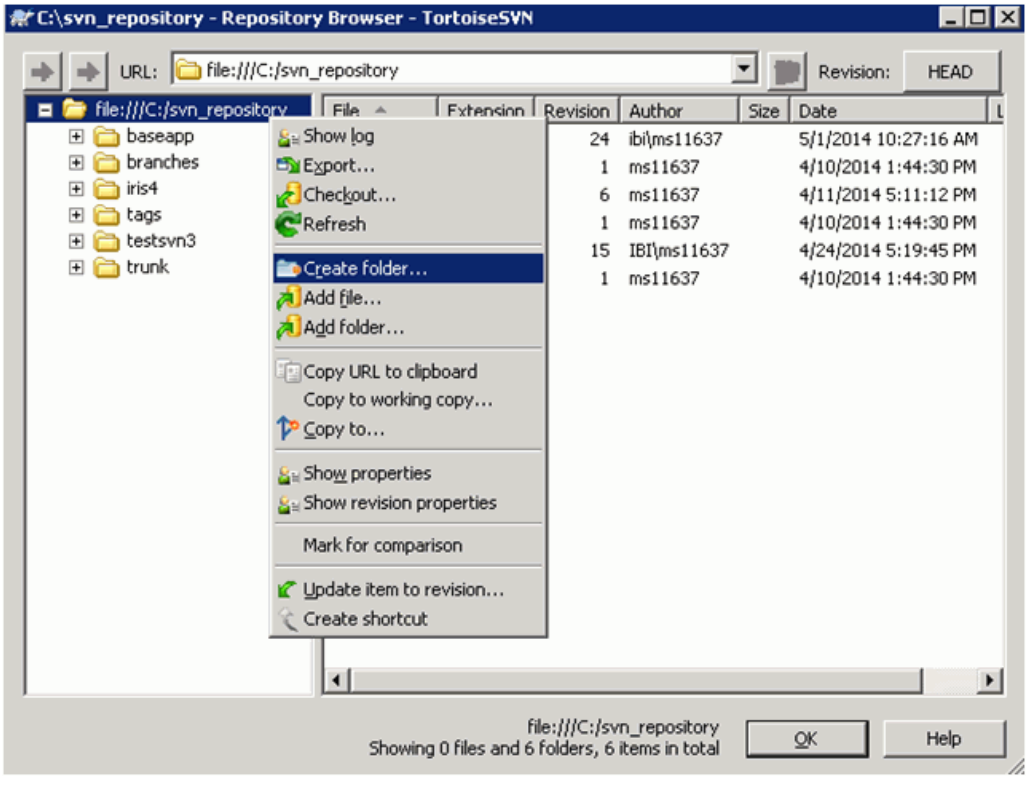
Procedure: How to Create Subversion Modules

Before you can create Subversion modules, you need to create a repository and add users. Once this is complete, you can create the modules through the Subversion command line, Subversion client, or at the Data Migrator configuration. The following procedure shows how to create a module with a Subversion client, TortoiseSVN. For details on how to create modules at Data Migrator configuration, see *How to Create Associations Between Application Directories and Subversion Modules* on page 738.

1. Install TortoiseSVN.
2. From Windows Explorer, right-click the *Subversion Repository* folder, point to *TortoiseSVN*, and then click *Repo-browser*.



3. Right-click on top of the tree path and click *Create folder*, as shown in the following image.



4. Enter the folder name. This should be the same as the Data Migrator Application Directory, such as ibisamp.

Procedure: How to Configure TIBCO Data Migrator to Work With Subversion

To use Subversion from client, you must install the client program. In Windows environment, this would be PushOk SVN SCC Proxy. Once this is installed and configured, the PushOk SVNSCC provider will show as an SCC Provider.

In the Data Management Console:

1. On the Home tab, in the Tools menu, click *Configure Source Control*.

   Note: If you have not properly configured PushOk SVNSCC Proxy, this option will not appear in the Tools group.

   The Configure Source Control dialog box opens.

2. Select *PushOk SVNSCC* as the SCC Provider from the drop-down menu.

Note: Selecting *none* removes source control from your system.

3. Optionally, click *Test* to make sure you can connect to the source control system from the DMC.
4. Click *OK*.

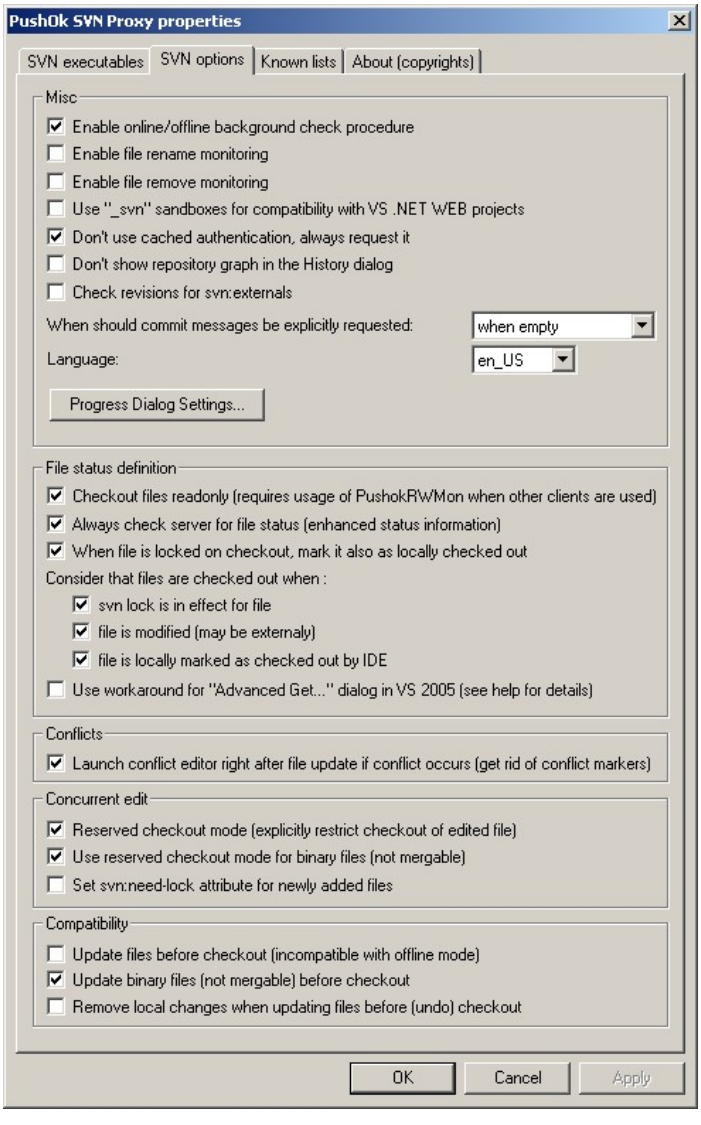
   The Set Source Control Provider dialog box opens.
5. Click *Yes*.

   Once an SCC Provider has been set, source control options appear on the right-click menus of the application directories.

In PushOk SVNSCC configuration plug-in:

1. Open the SVNSCC plug-in by going to *Start*, *All Programs*, *PushOk Software*, *SVNSCC*, and *Configure Plugin*.

2. Click the *SVN Options* tab and verify that the options are set the same, as shown in the following image.



Procedure: How to Create Associations Between Application Directories and Subversion Modules

Creating a source control folder provides a place to manage your Data Migrator files. Creating an association between your application directory and that folder tells Data Migrator where to manage those files.

Note: Developers must associate application directories inside their user home directory, while only the manager associates to the central application directory.

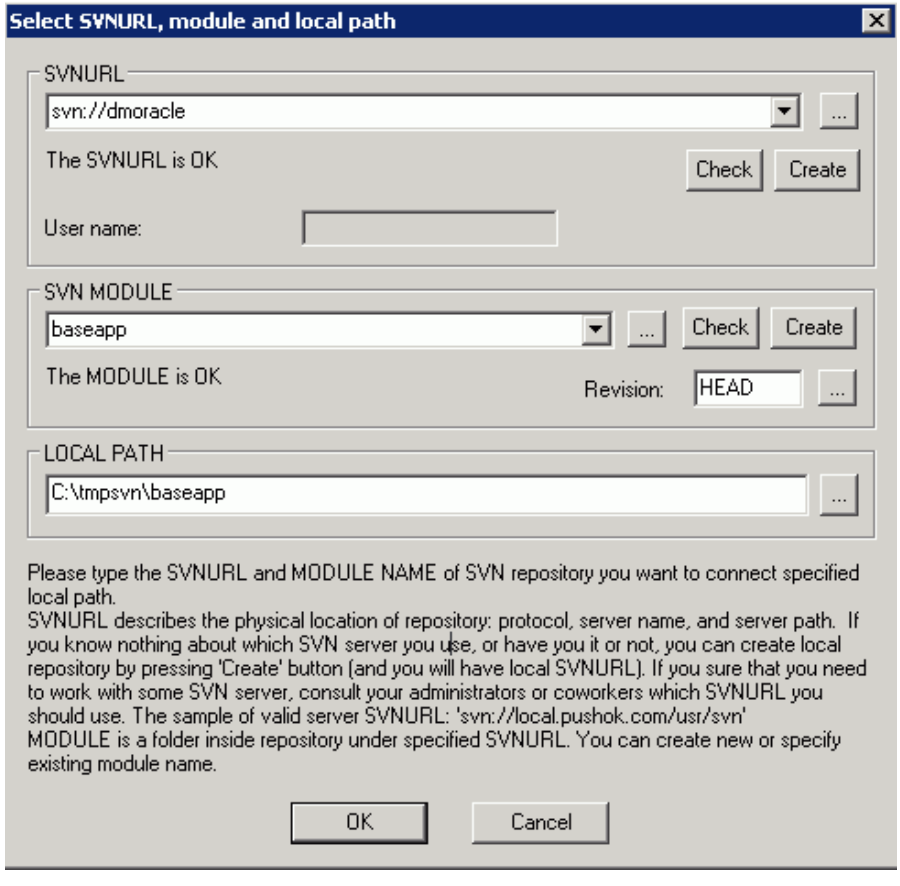
In the navigation pane of the Data Management Console:

1. Right-click an application directory, select *Source Control*, and then click *Create Association with Source Control*.

   The Associate Application with Source Control Project window opens.

2. Click *Associate*.

   The Select SVNURL, module and local path dialog box opens, as shown in the following image.



3. Enter the repository information in the SVNURL field. SVNURL describes the physical location of the repository. The repository is the directory that stores the revision copies of all the files.

The syntax for the SVNURL is:

`:protocol:pcname/path`

where:

protocol

Is the recommended protocol for a Windows server is svn.

pcname

Is the name of the machine running Subversion.

path

Is the location of the SVN repository. An example of a complete syntax is:

`svn://user1@PC1`.

Note: Each developer must connect with unique user name and password set in Subversion.

4. Click *Check*.

   If SVNURL is valid, the following message should appear:

   `The SVNURL is OK.`

5. Create a new module or specify the existing module name. Module is a specific directory (tree of directories) in the main repository, SVNURL. This should mimic the name of the Data Migrator application directory for consistency.
6. Click *Check*.

   If Module is invalid and does not exist, click *Create*. The module needs to be created only once by the developer, who is the first to associate the application directory with Subversion module. The rest of the developers need to connect to, and work with, the created or pre-existing module created by the first developer.

   If Module is valid, the following message should appear:

   `The MODULE is OK.`

7. The local path can be set to any local directory, such as C:\temp, but a different local path for each module must be specified. It is recommended that you create and keep the same module name inside a single local path, such as C:\temp\ibisamp.
8. Click *OK* to close the Select SVNURL, module and local path dialog box.
9. Click *OK* to close the Associate Application with Source Control Project dialog box. If association is successful, the Application Directory icon should display a lock to indicate the source control association.

You are now ready to use Subversion for source management.

Note: When you disconnect and reconnect with a different user to the DMC under source control, PushOk SVN Proxy might prompt you again for a user name and password to Subversion repository. If this occurs, login again with the appropriate Subversion user credentials.

Setup, Configure, and Associate Data Migrator to Work With Team Foundation Server (TFS)

Before you can create TFS Team Projects, you have to create a TFS Collection and setup users. A TFS Collection contains Team Projects, which can be associated to Data Migrator Application Directories. It is recommended that you use TFS Collections to differentiate various stages of the product life cycle, such as development, quality assurance, and production. This gives you the option to track different revisions across various deployment stages. However, the setup can vary depending on the requirements of the organization.

To use TFS for source control solution with a 32-bit version of DatamMigrator, Team Foundation Server MSSCCI Provider 2010 (or 2012) 32-bit is needed to convert the SCC API calls into TFS commands.

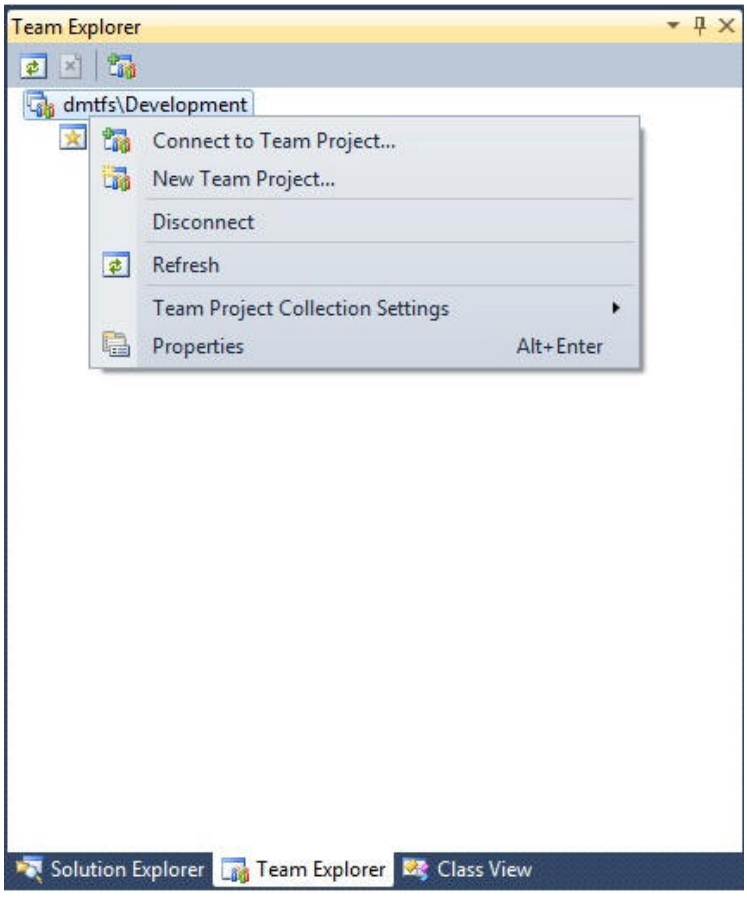
To use TFS for source control solution with a 64-bit version of Data Migrator, Team Foundation Server MSSCCI Provider 2012 (or higher) 64-bit is needed to convert the SCC API calls into TFS commands.

TFS MSSCCI Provider	Link Available:
2010 32-bit	*http://visualstudiogallery.msdn.microsoft.com/bce06506-be38-47a1-9f29-d3937d3d88d6*
2012 32-bit	*http://visualstudiogallery.msdn.microsoft.com/b5b5053e-af34-4fa3-9098-aaa3f3f007cd*
2012 64-bit	*https://marketplace.visualstudio.com/items?itemName=TFSPowerToolsTeam.MicrosoftVisualStudioTeamFoundationServer2012MSSCC-9956*
2013 & 2015 64-bit	*https://marketplace.visualstudio.com/items?itemName=TFSPowerToolsTeam.MicrosoftVisualStudioTeamFoundationServer20132015M-11387*

Procedure: How to Create TFS Collection and Team Projects

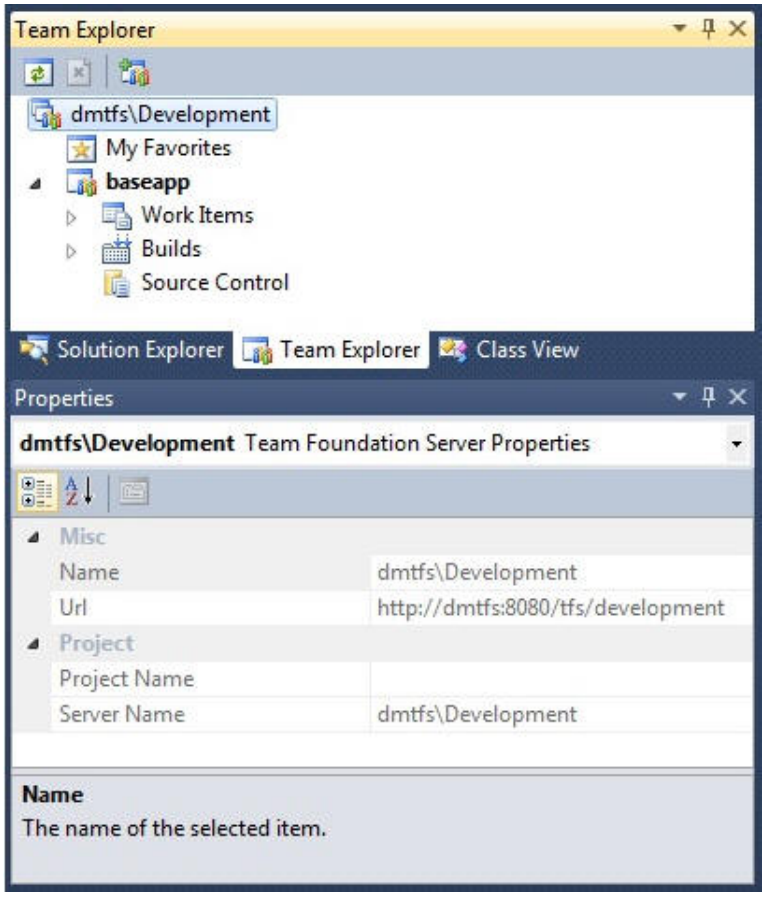
TFS Collections are created from the TFS Administration Console. Please contact your TFS administrator to have a collection created if you do not have one already set up.

TFS Team Projects are created through Visual Studio. You can create a new Team Project by right-clicking *TFS Collection* from the Team Explorer tab and clicking *New Team Project*, as shown in the following image.



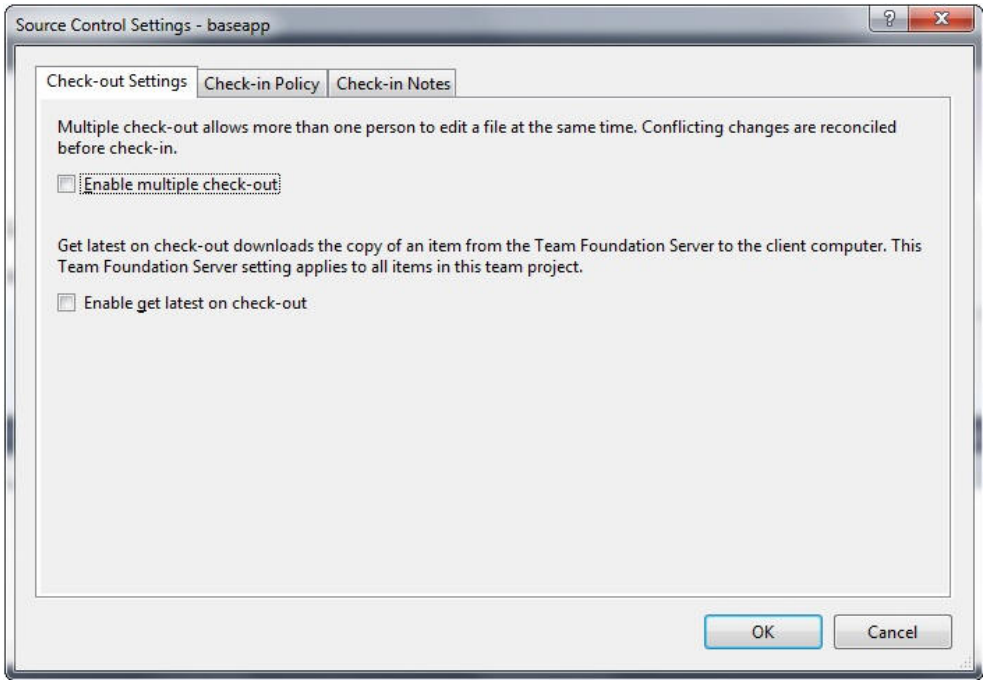
A wizard guides you in specifying the Team Project settings. The key setting is to name the team projects that correspond to the Data Migrator application directories. You can then click *Finish*. However, you can continue to the next steps if you want to specify the process template settings.

Use the information in the Properties pane to associate the TFS team project with Data Migrator application directories, as shown in the following image.



Procedure: How to Set an Exclusive Check-out Configuration

TFS allows concurrent check-out by default. It is required that you switch to an exclusive check-out method technique. To do this from Visual Studio, go to Team Explorer, right-click the Team Project, select *Team Project Settings*, and then click *Source Control*. Deselect the *Enable multiple check-out* check box and click *OK*, as shown in the following image.



Procedure: How to Configure TIBCO Data Migrator to Work With TFS

To use a source control system to source manage your Data Migrator flows, stored procedures, and synonyms, you must have TFS Collection and Team Projects set up and TFS MSSCCI Provider 2010 (or 2012) 32-bit installed on client machines.

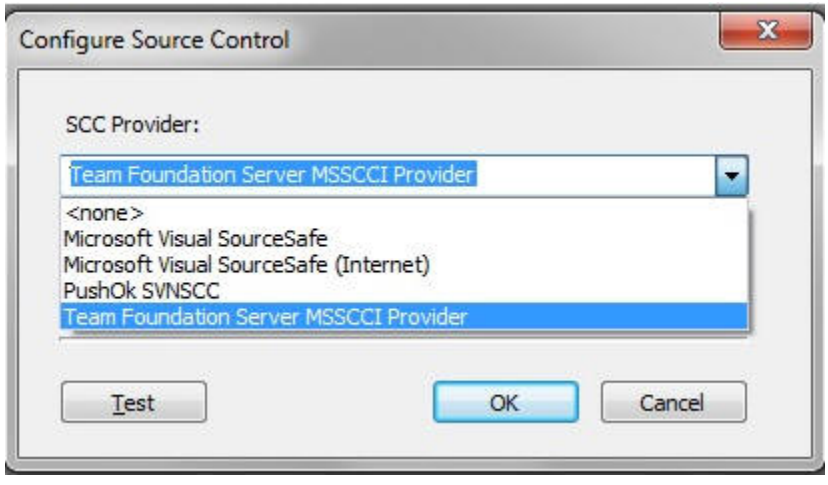
In the Data Management Console:

1. On the Home tab, in the Tools group, click *Configure Source Control*.

   Note: If you have not properly configured Team Foundation Server MSSCCI Provider, this option will not appear in the Tools group.

   The Configure Source Control dialog box opens.

2. Select *Team Foundation Server MSSCCI Provider* from the SSC Provider drop-down menu, as shown in the following image.



Note: Selecting *none* removes source control from your system.

3. Click *OK*.

   The Set Source Control Provider dialog box opens.

4. Click *Yes*.

   Once a SSC Provider has been set, source control options will appear in the context menus.

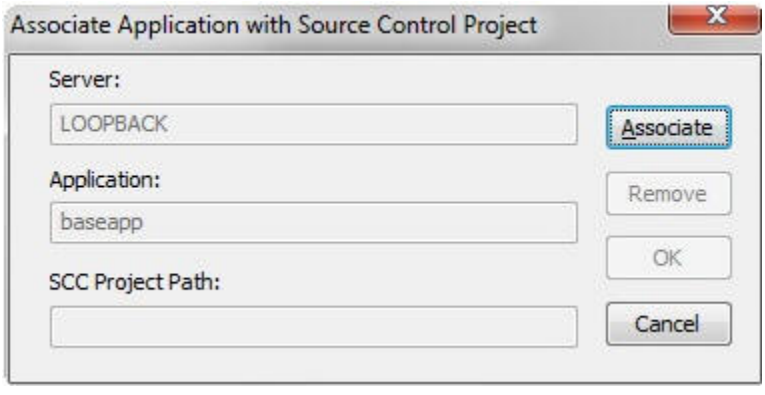
Procedure: How to Create Associations Between Application Directories and TFS Collection Projects

Creating a source control folder provides a place to manage your Data Migrator files. Creating an association between your application directory and that folder tells Data Migrator where to manage those files.

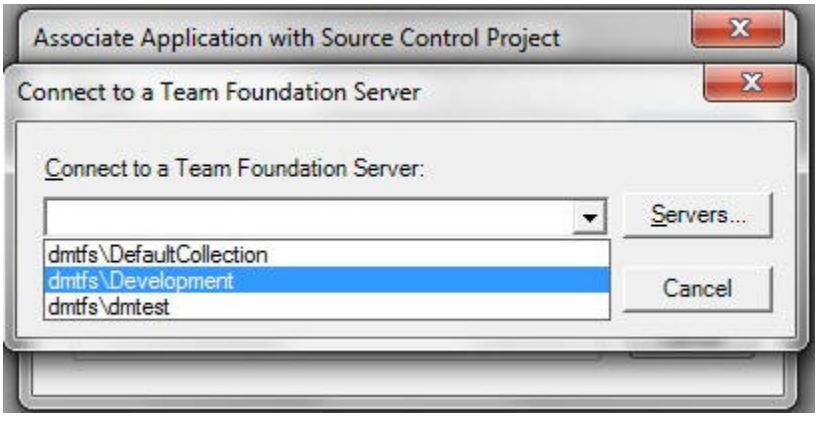
Note: Developers must associate application directories inside their user home directory, while only the manager associates to the central application directory.

In the Data Management Console:

1. Right-click an application directory in the navigation pane, select *Source Control*, and then click *Create Association with Source Control*. In this procedure, baseapp will be used as an example. The Associate Application with Source Control Project dialog box opens, as shown in the following image.



2. Click *Associate*.
3. Select the appropriate TFS Collection, as shown in the following image.



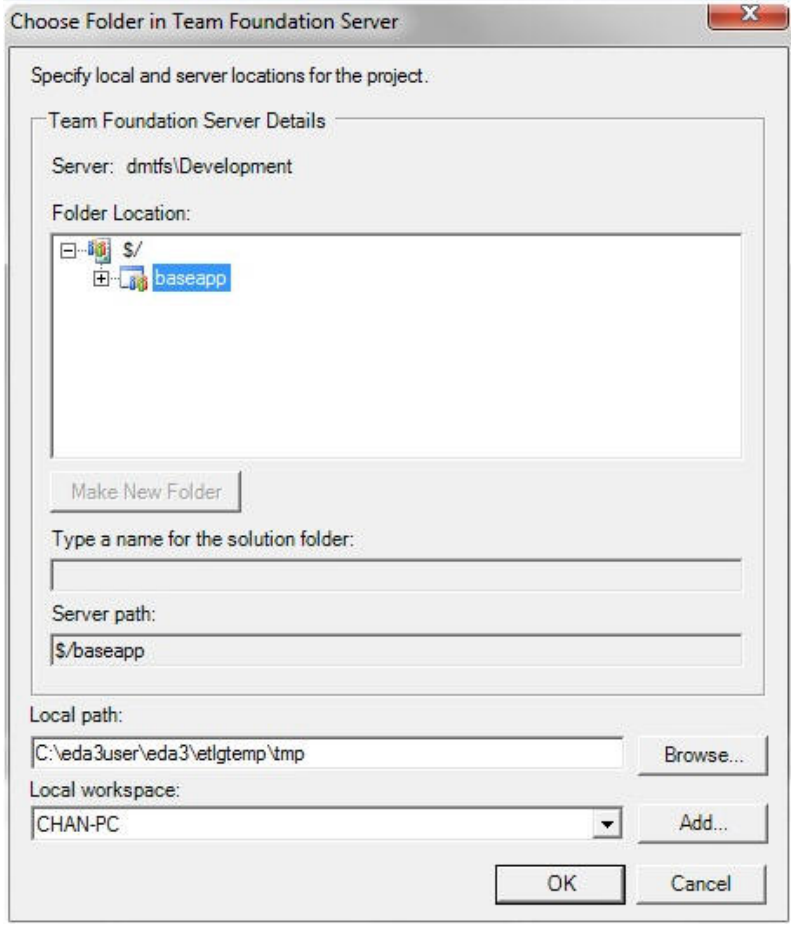
If the desired Team Foundation Server name is not listed in the drop-down list of the available Server names, you can perform the following steps to add the name:

a. Click the *Servers* button.

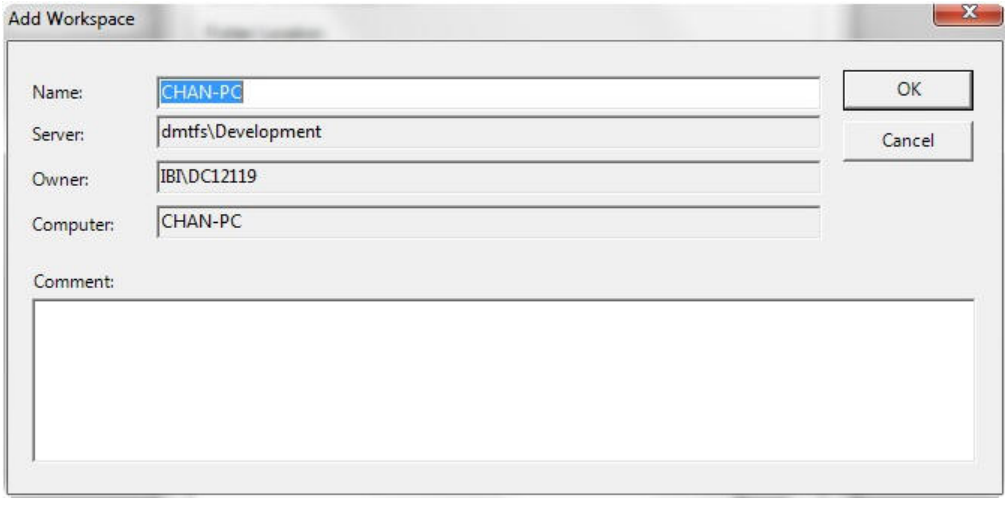
b. Click the *Add* button on the screen that opens next and enter the full URL of the Team Foundation Server name (for example, http://mydomain:8080/tfs/DefaultCollection).

c. Click the *OK* button.

4. Click *OK*.
5. You can now specify the local and server locations for the project. Specify the server path to match Team Projects. Specify the local path to the designated directory, which should be etlgtemp\tmp directory within your Server install path. For example, C:\ibi \srv82\DataMigrator\etlgtemp\server. The local path is where the source managed files temporarily reside.



6. You should use the default value for local workspace, which should be generated for you, as shown in the following image.



Note: The name of a temporary directory that is set for the Workspace on the Team Foundation Server is also used by Data Migrator as a 'Local path'.

7. Click *OK* to close the Associate Application with Source Control Project dialog box. The Application Directory icon displays a lock to indicate the source control association.

You are now ready to use TFS for source management.

Source Control Security Server and Client Setup

For source management solution, Data Migrator allows SCC control at the client side and stores home directories on the server side. Each developer will have a user home directory, which serves as a storage location for ongoing work. User home directories will be read-write for their owner with no access by any other developer, which prevents overwriting code among multiple developers. In addition, to prevent circumventing the source code, central application directories allow read-only privileges for all developers, while write access is granted only to the manager. The manager will create user home directories and deploy latest files from source control management system to the central application directories.

The descriptions and roles of these two positions are:

Developer

Create and edit flows. Their role is that of Application Administrator. By default, developers are assigned to Application Administrator roles. However, more or less privileges can be granted by the Server Administrator.

Manager

Create user home directories for all developers and deploy latest files from source control management system to the central application directories. Their role is that of Server Administrator. By default, the manager is assigned to Server Administrator. However, you can create different roles so that manager can deploy files, but cannot stop and start the server or change adapters.

Note: In the procedures below, Configure Server Security, Create and Configure User Home Directory, Register Users, and Assign Access Control are completed by the Manager. Log In Data Management Console with Security On and Associate User Home Directory with Source Control Project are completed by all Developers.

It is critical that you follow the procedures carefully and in their proper sequence.

Procedure: How to Configure Server Security

Before starting, the manager must set up server security so that developers are required to login.

1. From the DMC, expand the *Access Control* folder.
2. Right-click the *Security Providers* folder and click *Change Provider*.

   The Change Provider dialog box opens.
3. Set the Security Mode drop-down menu to the desired authentication type.
4. Enter your password and click *Apply and Stop Server*.
5. Restart the server.

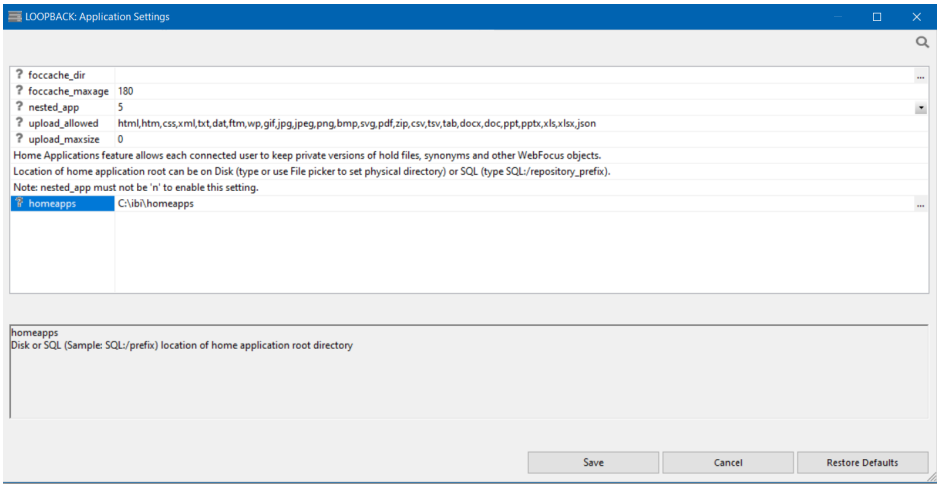
Procedure: How to Create and Configure User Home Directory

Once the manager logs in to the DMC with DM Security On, the next step is to create user home directory, also referred to as homeapps. Each developer will possess write access to their home directory inside homeapps.

1. From the DMC, right-click the *Workspace* folder, point to *Workspace SET*, and then click *Application Settings*.

   The Application Settings dialog box opens.
2. Enable nested_app by setting it to the *Y* value.

3. Enter the appropriate homeapps path. The homeapps path should be created under the Data Migrator root, ibi directory, as shown in the following image.



4. Click *Save*.

Procedure: How to Register Users

The manager registers new developers and assigns access roles from the Web Console.

1. From the DMC, expand the *Access Control* folder.
2. Expand the *Roles* folder, right-click *Application Administrator*, and then click *Register User*.
3. Enter the appropriate user or developer name and their credentials in the User Registration dialog box.

   Note: A user or developer name must be defined in Windows Local Users and Groups if using OPSYS authentication type on Windows. A similar rule would apply to another authentication type.

4. Click *Register*.

   An Option dialog box will appear to indicate that the new user will be registered. Click *OK*.

Procedure: How to Assign Access Control

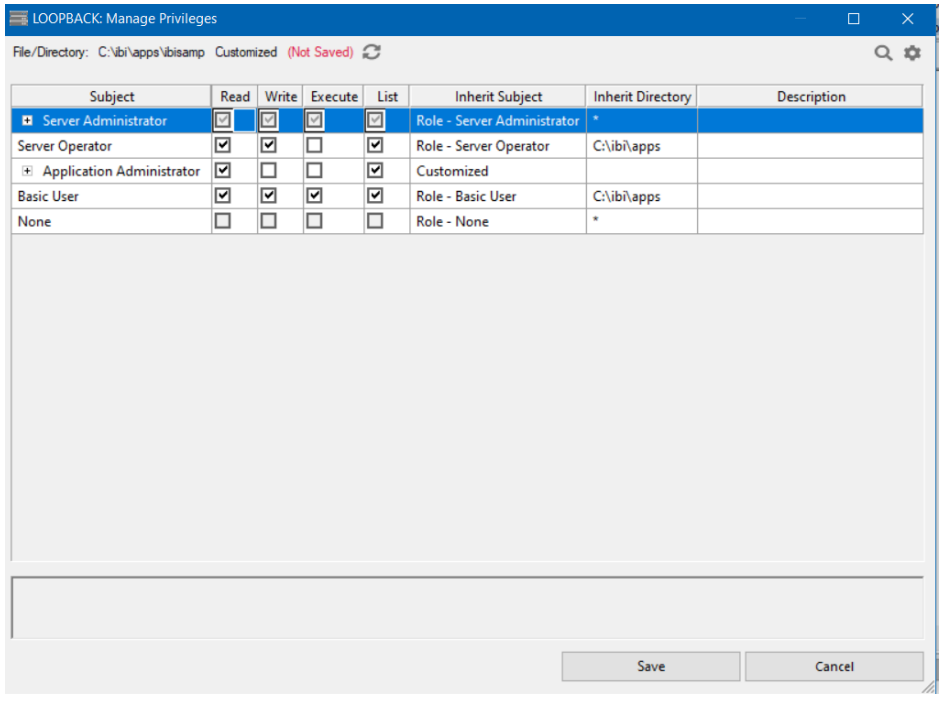
The default role setting for developers is that of Application Administrator. Each developer will possess write access to their home directory and read/list access to central application directories.

In the DMC:

1. Right-click an application directory and click *Privileges*.

   The File/Directory Privileges dialog box opens.

2. For the Application Administrator role, limit File Access options to *Read files* for AREAD and *List files* for ALIST, as shown in the following image.



3. Click *Save*.

Procedure: How to Log Onto Data Management Console With Security On

Developers log onto the Data Management Console with proper authentication.

1. Launch the Data Management Console.
2. Right-click a Server node and click *Properties*.
3. Log in with the appropriate User ID and Password that was created and registered by the manager. Click *OK*.
4. Right-click the Server node and click *Disconnect* and then *Connect* to connect through the new security setting.

Procedure: How to Associate User Home Directory With Source Control Project

Each developer will have a user home directory, which serves as a storage location for ongoing work. All developers must copy and paste the application directories into their user home directory.

1. In the DMC, right-click *Application Directories* and click *Create My Home*.

   My Home [developer name] directory appears.

2. Right-click the application directory you want to add to the source control, click *Copy*, and then paste it in the user home directory.

Repeat this step for all application directories you want to add to source control.

Using a Source Control System to Manage Procedures and Synonyms

Once you have created source control folders and created associations in the Data Management Console, the next step is to put the files under source control. In other words, you need to add them to a source control folder. You can add individual files or select a group of files from an application directory.

Files that have been added to a source control system need to be checked out before changes can be made. Checking files back in creates a history of their development.

Procedure: How to Add Files to a Source Control System

To track and manage changes, Data Migrator files must first be added to a source control system. This can be performed by developers or the manager. In either case, each new file needs to be added to the source control system just one time. Once the file has been added to the source control system, the rest of the developers can attain the copy from the source control system by invoking Get Latest Version.

Note: Files that are added or checked into a source control system become read-only until they are checked out.

In the navigation pane of the Data Management Console:

1. Right-click an application directory, multiselected objects, an individual flow, stored procedure, or synonym, select *Source Control*, and then click *Add/Check In*.

   The Add/Check In Files dialog box opens.

2. Select the files you want to add to source control. Files that are added are automatically checked in.

   Tip: You can use the *Select All* check box to both select and deselect all the files.

3. Optionally, you can check the files out by selecting the *Keep files checked out* check box.

4. Optionally, you can add comments by typing them into the *Comment* text box. If you are adding multiple files, they will all have the same comment.
5. Click *OK*.

   Note: The default listing for files in the Add/Check In dialog box is by Type. To list them by name, click the *Files* column heading.

   Icons display a lock to indicate that the files are under source control.

If you open your source control program, you will see that the files from the Data Migrator application directory have been added to the corresponding project folder. The synonym (dminv) has a .mas extension and the stored procedure (dmdefs) has a .fex extension. The flow (dmdemo) has two files stored in source control. The source code is stored in the file with the .fex extension. The file with the .etg extension holds the information required for the DMC interface.

Reference: Add/Check In Files Dialog Box

The Add/Check In Files dialog box is available by right-clicking an application directory, multi selected objects, an individual flow, stored procedure, or synonym, selecting *Source Control*, and then *Add/Check In*.

The Add/Check In Files dialog box has the following fields options:

Select All

Allows you to select or deselect all of the files in the application directory.

Files/Type/Action

Shows all of the files in the application directory. The files can be sorted by file name, type, or action.

Keep files checked out

Allows you to keep a file checked out that is being added or checked in. A source managed file must be checked out before you can make changes to it.

Comment

Allows you to add comments for the files.

Procedure: How to Get Latest Version From Source Control System

Once files have been added to source control, you can invoke Get Latest Version to attain a copy of the latest files from the source control system.

In the navigation pane of the Data Management Console:

1. Right-click an application directory, multi selected objects, an individual flow, stored procedure, or synonym, select *Source Control*, and then click *Get Latest Version*.

   The Get Latest Version window will appear.

2. Click *OK*.

   Note: If you invoke the Get Latest Version from application directory, the Get Latest Version dialog box will list all the files that have been added to the source control system. There is a Select All option to get all the files from source control system, or you can optionally select the individual files that you want.

Procedure: How to Check Files Out of Source Control

For files that are being source managed, you must check them out before you can make and save changes. Until the files are checked out, they remain read-only.

In the navigation pane of the Data Management Console:

1. Right-click an application directory, multi selected objects, an individual flow, stored procedure, or synonym, select *Source Control*, and then click *Check Out*.

   The Check Out dialog box opens.

2. Select the files you want to check out.

   Tip: You can use the Select All check box to both select and deselect all the files.

3. Optionally, add comments by typing them into the Comment text box. If you are checking out multiple files, they will all have the same comment.

4. Click *OK*.

   Icons in the application directories display a red check mark to indicate that the files have been checked out.

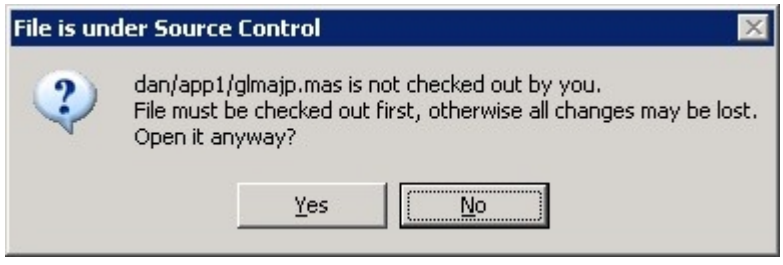
   Note: The default listing for files in the Check Out dialog box is by Type. To list them by name, click the *Files* column heading.

   If you open the source control program, you will see that the files that you checked out from the Data Management Console have been checked out of the source control program. These files are no longer read-only, which allows you to make changes before checking them back in to source control.

   If a data source has an Access File associated with it, the Access File will also be automatically checked out, in the same way that an .etg file is checked in or out with its associated flow.

Note:

- If you try to edit a file that has been added to source management without checking it out, the following message appears:



- If you try to check out a source-managed file that has already been checked out by another user, the following message appears:



Reference: Check Out Files Dialog Box

The Check Out Files dialog box is available by right-clicking an application directory, multi selected objects, an individual flow, stored procedure, or synonym, selecting *Source Control*, and then clicking *Check Out*.

The Check Out Files dialog box has the following fields and options:

Select All

Allows you to select or deselect all of the files that have been checked in and/or added to the application directory.

Files/Type/Action

Shows all of the files that have been checked in or added to the application directory. The files can be sorted by file name, type, or action.

Comment

Allows you to add comments for the files.

Procedure: How to Check Files Into Source Control

After you have made changes to files, you must check them back into your source control system to maintain a record of those changes. Files that are checked in are copied back to the server.

In the navigation pane of the Data Management Console:

1. Right-click an application directory, multi selected objects, an individual flow, stored procedure, or synonym, select *Source Control*, and then click *Add/Check In*.

   The Add/Check In dialog box opens.

2. Select the files you want to check in. Only files that have been checked out have Check In as the possible action. The action for files that have not been added to source control is Add File.

   Tip: You can use the Select All check box to both select and deselect all the files.

3. Optionally, you can keep files checked out by selecting the *Keep files checked out* check box.
4. Optionally, you can add comments by typing them into the Comment text box. If you are checking in multiple files, they will all have the same comment.
5. Click *OK*.

   Note: The default listing for files in the Add/Check In dialog box is by Type. To list them by name or action, click the *Files* or *Action* column headings.

Procedure: How to Undo a Check Out From Source Control

If you decide that you do not want to make changes to a file, you can undo its check out from the source control system. The file becomes read-only again, and source control maintains the version as it existed before the file was checked out.

In the navigation pane of the Data Management Console:

1. Right-click an application directory, multi selected objects, an individual flow, stored procedure, or synonym, select *Source Control*, and then click *Undo Check Out*.

   The Undo Check Out dialog box opens.

2. Select the files whose check out you want to undo.

   Tip: You can use the Select All check box to both select and deselect all the files.

3. Click *OK*.

   Note: The default listing for files in the Undo Check Out dialog box is by file name. To list them by Type, click the *Type* column heading.

Reference: Undo Check Out Dialog Box

The Undo Check Out Files dialog box is available by right-clicking an application directory, multi selected objects, an individual flow, stored procedure, or synonym, selecting *Source Control*, and then clicking *Undo Check Out*.

The Undo Check Out Files dialog box has the following fields and options:

Select All

Allows you to select or deselect all of the files that have been checked out.

Files/Type/Action

Shows all of the files that have been checked out from the application directory. The files can be sorted by file name or type.

Procedure: How to View the History, Versions, and Differences of a File

The source management history records of a file each date and time it was checked in, as well as the user involved.

In the navigation pane of the Data Management Console:

1. Right-click an individual flow, stored procedure, or synonym, select *Source Control*, and then click *History*.

   The History dialog box opens, showing the flow, stored procedure, or synonym.

2. Click *OK*.

   The History Options dialog box opens.

3. Optionally, use the *Include Labels* or *Labels Only* check boxes to show labeled activity.

   Note: Labels can be useful to distinguish versions of a particular file. You can assign labels from the History Details dialog box that is available from the Data Management Console History of dialog box.

4. Optionally, use the From and To fields to specify a date range for the history.
5. Optionally, use the User field to specify the history for a specific users activity.

   The History of dialog box opens, showing the user, date, and time for each check in.

Procedure: How to Deploy Latest Files From Source Control to Central Application Directory

The manager can deploy the latest files from source control to central application directory. Since the working directory of a manager is the central application directory, all they need to do is invoke Get Latest Version from each application directory or an individual flow, stored procedure, or synonym.

In the navigation pane of the Data Management Console:

1. Right-click an application directory, multi selected objects, an individual flow, stored procedure, or synonym, select *Source Control*, and then click *Get Latest Version*.

   The Get Latest Version dialog box will open.

2. Click *OK*.

   Note: If you invoke the Get Latest Version from application directory, the Get Latest Version dialog box will list all the files that have been added to the source control system. There is also a Select All option which gets all the files from the source control system.

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, FOCUS, iWay, Omni-Gen, Omni-HealthData, and WebFOCUS are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.

Copyright © 2021. TIBCO Software Inc. All Rights Reserved.

Index

-TYPE command *703*

&&CM__AUTHOR *315*, *387*

&&CM__REQUEST *315*, *388*

&&CM__RETURN *315*

&&CM__TARGET *315*, *388*

&&KILL_RPC *316*, *388*

&ACCEPTS *318*, *388*

&CHNGD *318*, *388*

&DATE *316*

&DBMSERR *318*, *389*

&DELTD *318*, *389*

&DMY *317*

&DUPLS *318*, *389*

&EXITRC *318*

&FORMAT *318*, *389*

&INPUT *318*, *389*

&INVALID *318*, *389*

&LINES *319*, *389*

&MDY *317*

&NOMATCH *318*, *389*

&RECORDS *319*, *390*

&REJECTS *318*, *389*

&TOD *317*

&TRANS *318*, *389*

&YMD *317*

A

Access File Text View tab *497*

adding data adapters *48*

adding email in a process flow *394*

adding IDs *38*

adding nodes *417*

Aerial View *133*

aggregating columns *225*

alphanumeric fields *345*, *351*

APP PATH *46*

- configuring *46*

application button *105*

application directories *43*, *44*, *46*

- adding *45*
- creating *43*, *419*
- deleting *44*, *45*

application menu *105*

application paths *42*, *46*

- APP PATH *46*
- application path *42*
- configuring *42*
- restricting access *42*, *98*, *100*

applications *87*

Assist button *308*, *580*

automapping *305–307*

automatic passthru (APT) *476*, *478*

- creating pre-extract procedures *478*
- real numbers and *478*
- requirements *476*

B

BLOB format *344*
branching in process flows *401*, *447*
building SQL calculations *434*
bulk load options *262*, *264*, *265*
    Db2 *282*
    Nucleus *284*
    Oracle *267*
    Teradata *286*

C

Change Data Capture *635*
    configuration *656*
    flows *667*
    logs *665*
    tutorial *659*
ChangeMan DS *727*
changing join types *428*
ClearCase *729*, *730*
CMASAP procedure *693*
    executing flows *693*
CMRUN procedure *681*
    executing flows *681*, *682*, *685–689*
    return codes *681*, *690*
code *371*
column management *159*
columns *220*
    aggregating *225*
    changing display *127*
    changing display in the Synonym Editor *129*
columns *220*
    extract transformations *226*
    information displayed *129*
    managing *428*
    mapping *436*
    naming *89*, *97*
    Selected Columns window *221*, *223*
    selecting *220*, *221*, *428*
    text *88*
    variable character length *88*
    virtual *424*
commands *242*
    -TYPE *703*
complex data flows *175*
complex mapping *305*, *309*
components *178*
    adding *178*
    connecting *181*
    deleting *178*, *181*
    disconnecting *181*, *182*
Concurrent Versions System (CVS) *731*
configuring application paths *42*
configuring Data Migrator *27*
configuring email notification *65*
configuring server nodes *417*
configuring SMTP services *65*
connecting objects *181*, *395*, *397*, *403*
connection logic *395*, *397*, *403*, *449*
    specifying *449*
Connector messages *700*

connector objects *373*, *395*
Console Log *142*
copying *123*
    procedures *123*
    synonyms *123*
Core Engine Settings *707*
creating application directories *419*
creating branches *401*
creating joins *209*
creating metadata *54*
creating objects *370*
creating pre-extract procedures *478*
creating process flows *376*, *377*
    from data flows *378*
creating stored procedures
    in Data Management Console *381*
creating synonyms *54*
CVS (Concurrent Versions System) *732*

D

data adapters *48*
    adding *48*
data extraction *475*
Data Flow object *372*
Data Flow tab *21*, *124*, *419*
data flows *21*, *171*, *172*, *314*, *419*
    adding objects *178*
    complex *175*
    creating *177*, *415*, *419*
    Data Migrator variables *314*
data flows *21*, *171*, *172*, *314*, *419*
    deleting objects *178*
    designing *21*
    direct load *363*
    interface *172*
    objects *175*
    running *444*
    simple *174*
    storing *114*
    system variables *314*
    using variables *314*
    validating *125*
data lineage *538*
data loading *479*
data management *14*
Data Management Console *15*, *17–19*, *32*, *87*, *183*
    adding application directories *45*
    creating application directories *43*
    deleting application directories *44*, *45*
    designing process flows *21*
    filtering *245*, *319*
    functions *17*
    interface *103*
    launching Web Console *38*
    navigation pane *18*, *19*, *110*
    ribbon *107*
    server administration tasks *38*
    sessions *169*
    shortcuts *109*

Data Management Console *15*, *17*–*19*, *32*, *87*, *183*
sorting *251*
source component *183*
SQLAnswerSet component *242*
style menu *108*
targets *254*, *261*, *305*–*307*, *309*
testing connections *37*
testing servers *37*
Text View tab *21*
workspace *123*
data marts *14*
data migration *14*
Data Migrator *13*, *14*
architecture *15*
components *15*
configuring *27*
creating data flows *415*
creating process flows *415*
help *26*
installing *27*
processing *15*
security *98*
upgrading *82*
data objects *446*
adding to process flows *446*
data profiling *119*, *530*, *533*
in synonyms *530*
outliers *533*
patterns *533*
data profiling *119*, *530*, *533*
single column *533*
statistics *533*, *536*, *537*
values *533*
data source columns *189*
updating *190*
data sources *14*, *88*, *93*, *179*, *183*
accessing *92*
column names *89*
columns *220*
joining *204*, *420*
naming *89*
performance *92*
reviewing *92*
selecting *183*, *420*
updating *189*
data store tables *669*
ETLSTATS *675*
data targets *14*, *88*, *94*
accessing *98*
adding *253*, *254*, *436*
column names *89*, *97*
creating *441*
existing *254*, *261*
fixed format files *254*
flat files *254*
guidelines *97*
local relational *95*
naming *97*
new *254*

data targets *14*, *88*, *94*
non-relational *96*, *254*
performance *94*
relational *254*
remote destinations *97*
reviewing *94*
security *98*
selecting *254*, *261*
Server Transfer Files *254*
data warehousing *14*
designing applications *87*
data
data sources *93*
data targets *94*
filtering *433*
validating *440*
DB log parameters *656*
DB_LOOKUP *308*
Db2 *282*
tables *664*
DBMS errors
limit *706*
DBMS
SQL flows *365*
DEFINE *527*
deleting
procedures *123*
synonyms *123*
delimited file targets *255*
designing data flows *21*
designing process flows *21*
detailed flow reports *470*, *471*
Dialogue Manager commands
-TYPE *703*
direct load flows *363*, *663*
creating *363*

E

e-mail objects *373*
editing conditions *397*, *403*
editing SQL commands *242*
editing synonyms *120*
email notification *65*, *71*
properties *147*
email objects *394*
error messages *697*
resolutions *702*
run-time *701*
Server *698*
errors *697*
ETLLOG table
Server Log *673*
ETLSTATS table *675*
Server Statistics table *673*
event-based scheduling *681*
CMASAP procedure *693*
CMRUN procedure *681*, *682*, *685–689*
FOCUS and *695*
status *695*
WebFOCUS and *695*

executing flows *381*
procedures *381*
simultaneously *401*, *402*
expressions *203*
functions and *325*
order of evaluation *203*
extract transformations *190*, *191*, *424*
calculations *192*, *194*, *226*, *227*
constant values *226*
creating *190*–*192*, *194*, *226*, *227*, *424*
date values *226*
in data source object *191*
SQL Calculator *192*, *194*, *226*–*228*
SQL functions *230*
time values *226*
extracting data *475*

F

field formats *343*, *345*
alphanumeric fields *345*, *351*
formatting new fields *345*, *351*
Field View tab *486*
Filter Calculator *249*
filtering *245*, *319*, *433*
data *433*
expressions *434*
Filter Calculator *245*, *249*
fixed format files *254*
flat files *254*
floating-point numbers *478*
flow object *372*
flow properties *356*
flow reports *470*
detailed *470*, *471*
summary *452*, *456*
Flow tab *136*
flows *14*, *21*, *98*, *473*
execution properties *145*
building *415*, *416*
controlling *393*
copying *123*
creating *415*, *416*, *420*
data sources *183*, *420*
data targets *254*, *261*
email notification *147*
field formats *343*, *345*
filtering *245*, *319*
general properties *145*
graphical representation *473*
grouping *402*
mappings *305*
printing *473*
procedures *393*
properties *144*, *356*
record logging properties *150*
report *470*
restart properties *147*
running *98*
running remotely *681*
scheduling *407*, *408*, *681*

flows *14*, *21*, *98*, *473*
    sorting *251*
    status *695*
    testing SQL *242*
FOCUS *695*
    executing flows *695*
functions *325*
    changing display *128*

G

global variables *387*
graphical representation of flows *473*

H

help button *109*
Home tab *133*

I

ibisamp directory *42*
Impact Analysis *119*
inner joins *426*
inserting sample data *121*
installing Client
    Install Wizard *33*
installing Data Migrator *27*, *32*

J

Join Calculator *204*, *209*, *211*
Join Properties window *209*, *428*
joining data sources *204*, *420*, *426*
    Join Calculator *204*
    Join Properties window *209*
    performance considerations *204*, *205*
joins *204*, *420*
    assigning properties *426*
    inner *426*
    types *428*

L

List View tab *488*
listening parameters *657*
load options *262*, *290*, *291*
    targets *290*, *291*, *295*, *302*, *303*
local relational data targets *95*
log file *451*, *472*
    viewing *470*, *472*
log messages *375*
log reports *451*, *454*
    columns *452*
logs *114*
    sample data *668*
lookup cache *481*
lookups *579*

M

managing columns *428*
mappings *305–307*, *309*, *311*
    automapping *305–307*
    columns *436*

mappings *305–307*, *309*, *311*
complex mapping *305*, *309*
field formats *343*, *345*
simple mapping *305*
temporary columns *305*
testing *309*
Message window *19*
messages *142*
run-time *703*
metadata *54*
creating *54*
testing *54*, *58*
Modeling View tab *495*, *523*
multiple data targets *441*

N

navigation pane *18*, *370*
non-automatic passthru *477*
non-relational data targets *96*, *254*
Nucleus *284*
NULLABLE *706*

O

objects *178*, *370*
process flows *371*
Oracle *267*
Oracle tables *664*
order of evaluation *203*

P

Parallel Group object *372*, *401*, *402*
parameters *705*
setting *705*
passwords *69*, *692*
supplying with procedures *692*, *693*
performance
data loading *479*
extracting data *475*
Print Preview tab *469*
printing flows *473*
printing
synonyms *496*
procedures *114*, *380*, *391*
copying *123*
deleting *123*
executing *393*
flows *393*
process flows *380*
supplying passwords *692*, *693*
Process Flow tab *21*, *129*, *370*
process flows *21*, *369*, *370*, *445*
adding components *375*
adding data objects *446*
adding email objects *394*
adding procedures *380*, *391*
adding stored procedures *380*
branching *401*, *447*
complex *375*
creating *376*, *415*

process flows *21*, *369*, *370*, *445*
creating from data flows *378*
creating new *377*
designing *21*
objects *369*, *371*
running *406*
scheduling *406–408*
simple *374*
standalone procedures *377*
storing *114*
properties
Application Directory *155*
email notification *147*
execution *145*
flows *144*
general *145*
record logging *150*
restart *147*
prototype synonym *519*

Q

Quick Access Toolbar *106*

R

read limits *657*
real numbers *478*, *481*
automatic passthru and *478*, *481*
changing formats *478*, *481*
record retrieval *434*
recreating DBMS tables *120*
relational data targets *254*
relational lookups *308*
relational targets *261*, *290*, *293*
remote destination data targets *97*
remote servers *30*
adding *477*
remote table access *97*
reports *114*, *451*, *456*
copying *465*, *466*
flow *452*, *456*, *470*, *471*
fonts *466*
log *451*, *454*
Report tab *140*
running *457*, *458*, *461*
statistics *452*
return codes *681*
CMRUN procedure *681*, *690*
ribbon *107*, *133*
Metadata tab *497*
tabs *108*
running on recurring schedule *406*, *407*
running on specific days *406*, *408*
running once *406*, *407*
running process flows *375*, *406*

S

sample data *58*, *118*, *184*, *188*
inserting *121*
Scheduler *66*, *411*
Scheduler configuration page *67*

Scheduler window *411*
scheduling flows *406*–*408*, *681*
event-based scheduling *681*, *682*, *685*–*689*, *693*, *695*
Scheduler *407*, *408*
scheduling process flows *375*, *407*
security *98*
Segment View tab *487*
Select Synonym dialog box *179*
Selected Columns window *221*, *223*
selecting columns *220*, *221*, *223*, *428*
Server *28*
administration *38*
connecting *419*
properties *152*
run-time errors *701*
Server Node Configuration window *153*
setup *28*
SET command *705*
Set variables object *373*
setting Wait conditions *403*
simple data flows *174*
simple mapping *305*
slowly changing dimensions *507*
SMTP services *65*
Sort Columns window *252*
sorting *251*, *252*
source component *183*
source control *752*
adding files *752*, *753*
source control *752*
checking in files *756*
checking out files *754*, *755*
ClearCase *730*
CVS *732*
history *757*
undo check out *756*, *757*
source management *709*
ChangeMan DS *727*
ClearCase *729*
Concurrent Versions System (CVS) *731*
configuring *726*
SCC control *748*
source control *752*
Subversion *734*
Team Foundation Server (TFS) *741*
source window *183*
SQL Calculator *192*, *194*, *226*–*228*
SQL commands *242*
editing *242*
testing *242*
SQL functions *230*, *231*
SQL operators *230*
SQL Server
tables *637*, *664*
SQLAnswerSet component *242*
standalone process flows *377*
standalone stored procedures *381*
Start object *371*
statistical variables *318*, *388*

statistics reports *452*
columns *453*
Stored procedure object *373*
stored procedures *447*
adding to process flows *380*
creating *381*, *447*
executing *375*
storing *114*
Subversion *734*
summary flow reports *452*, *456*
Synonym Editor *485*
Segment shortcut menu *488*
Access File Text View tab *497*
changing column display *129*
Column shortcut menu *491*
Field View tab *486*
file attributes summary *501*
File Listener attributes *514*
List View tab *488*
Metadata tab *497*
Modeling View tab *495*, *523*
segment attributes summary *502*
Segment View tab *487*
synonym attributes *499*
Text View tab *496*
virtual columns *527*, *528*
synonyms *19*, *54*, *420*
copying *123*
creating *54*
creating using Modeling View *523*
synonyms *19*, *54*, *420*
data lineage *538*
data profiling *530*
data sources and *88*
deleting *123*
editing *120*
guidelines *88*, *89*
indexes *526*
key analysis *531*
log tables *660*
migrating *82*
printing *496*
prototype file *519*
testing *54*, *58*
using variables *520*
virtual columns *527*, *528*
system variables *316*

T

tabs *108*
Metadata *497*
Data Flow *124*
flow *136*
Home *133*
print preview *469*
Process Flow *129*
report *140*
text editor *142*
Text View *130*
user functions *139*

targets *254*, *261*, *290*, *291*, *299*, *305*
fixed format files *302*
flat files *299*, *302*
FOCUS files *291*, *295*
formatted files *303*
Fusion files *291*, *295*
relational *254*, *261*, *290*, *293*
Transformations dialog box *311*
Team Foundation Server (TFS) *741*
temporary columns *305*
Teradata *286*
Test Transforms tool *311*, *426*
testing flows *64*
testing metadata *58*
testing SQL *242*
testing synonyms *58*
text columns *88*, *92*
text editor *131*
Text Editor tab *142*
Text View tab *130*, *371*, *496*
trace *703*
tracking results *447*
Transformation Calculator *203*, *425*
functions *325*
order of evaluation *203*
Transformations dialog box *311*
troubleshooting *697*
server messages *698*

U

updating passwords *69*
upgrading Data Migrator *82*
User Functions tab *139*
user preferences *156*
user profiles *42*
using the Transformation Calculator *425*

V

validating data *440*
VARCHAR *88*
variable length character columns *88*
variables *314*, *315*, *319*, *387*
changing display *127*
statistical *318*, *388*
system *316*, *319*
verifying data *423*
version control
adding files *752*, *753*
checking in files *756*
checking out files *754*, *755*
ClearCase *730*
CVS *732*
history *757*
undo check out *756*, *757*
viewing code *371*
viewing flow reports *470*, *471*
virtual columns *424*
adding *527*
creating *528*

W

Wait conditions *403*
Wait object *372*, *401*, *402*
Web Console *38*
    accessing *38*
Web Console *38*
    launching *38*
    Scheduler configuration page *67*
WebFOCUS *695*
    executing flows *695*