

IBM® DB2 Universal Database™



Data Warehouse Center Administration Guide

Version 8.2

IBM® DB2 Universal Database™



Data Warehouse Center Administration Guide

Version 8.2

Before using this information and the product it supports, be sure to read the general information under *Notices*.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

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

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1996, 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book ix

Who should read this book ix

Prerequisite publications ix

Chapter 1. About data warehousing . . . 1

What solutions does data warehousing provide? . . . 1

Warehouse tasks 2

Data warehouse objects 2

Subject areas 2

Warehouse sources 3

Warehouse targets 3

Warehouse control databases 3

Warehouse agents and agent sites 3

Processes and steps 4

Chapter 2. Setting up your warehouse. . 7

Starting the Data Warehouse Center 7

Starting the Data Warehouse Center administrative interface 8

Starting and stopping the Data Warehouse Center server and logger 9

Starting and stopping the warehouse server and logger (Windows). 9

Starting and stopping the warehouse server and logger (AIX) 9

Verifying that the warehouse server and logger daemons are running (AIX) 10

Starting the warehouse agent daemon 11

Warehouse agent daemons 11

Connectivity requirements for the warehouse server and the warehouse agent 11

Starting the warehouse agent daemon (Windows) 12

Starting the iSeries warehouse agent daemon . . 12

Verifying that the iSeries warehouse agent started 12

Verifying that the iSeries warehouse agent daemon is still running 12

Configuring TCP/IP on z/OS 13

Starting the zSeries agent daemon in the foreground 13

Starting the zSeries warehouse agent daemon in the background 14

Verifying that the zSeries warehouse agent is running. 14

Verifying communication between the warehouse server and the warehouse agent 15

Stopping the warehouse agent daemon 15

Stopping the warehouse agent daemon (Windows). 15

Stopping the warehouse agent daemon (AIX, Solaris Operating Environment, Linux) 16

Stopping the iSeries warehouse agent daemon. 16

Stopping the warehouse agent daemon (zSeries) 16

Defining agent sites 17

Agent site configurations 17

Defining an agent site 18

Updating your environment variables on z/OS 19

Data Warehouse Center security 20

Defining warehouse security. 22

Chapter 3. Setting up DB2 warehouse sources 25

Supported DB2 data sources. 25

Warehouse agent support for DB2 sources 26

Setting up connectivity for DB2 sources (Windows NT, Windows 2000, Windows XP) 26

Setting up connectivity for DB2 Universal Database databases (Windows) 26

Setting up connectivity for DB2 DRDA databases (Windows). 26

Setting up connectivity for DB2 sources (AIX) . . . 27

Setting up connectivity for a DB2 Universal Database source (AIX) 27

Setting up connectivity for a DB2 DRDA database source (AIX) 28

Setting up connectivity for DB2 sources (Solaris Operating Environment, Linux). 28

Setting up connectivity for a DB2 Universal Database source (Solaris Operating Environment, Linux) 28

Setting up connectivity for a DRDA data source (Solaris Operating Environment, Linux) 29

Setting up connectivity for DB2 sources (OS/2) . . 30

Setting up connectivity for a DB2 Universal Database source (OS/2) 30

Setting up connectivity for a DRDA data source (OS/2) 30

Accessing source databases 30

Defining privileges to DB2 data sources 31

Establishing connectivity to DB2 data sources . . 31

Setting up access to DB2 DRDA data sources . . 31

Defining privileges to DB2 DRDA data sources 32

Setting up a DB2 Connect gateway site (Windows). 33

Connecting to DB2 DRDA data sources 33

Connectivity requirements for remote databases . . 34

Connectivity requirements for remote databases from the iSeries warehouse agent 34

Setting up access to local and remote databases from the iSeries warehouse agent 35

Example of how to catalog local and remote database names for the iSeries warehouse agent . 36

Viewing and editing remote relational database directory entries 36

Requirements for accessing remote data sources with the zSeries warehouse agent 36

Requirements for accessing relational data sources with the zSeries warehouse agent . . . 37

DB2 warehouse sources 37

Ordinary identifiers 37

Delimited identifiers 37

Metadata from tables	37
Agent sites	38
Defining DB2 warehouse sources	38

Chapter 4. Setting up non-DB2 warehouse sources 39

Supported non-DB2 data sources	39
Warehouse agent support for non-DB2 sources	40
Column size truncation for non-DB2 warehouse sources	40
Setting up connectivity for non-DB2 sources (Windows).	41
Setting up connectivity for a Sybase Adaptive Server source – without client (Windows)	41
Setting up connectivity for an Oracle source (Windows).	41
Setting up connectivity for an Informix 9.2 source – with client (Windows)	42
Setting up connectivity for an Informix 9.2 source – without client (Windows)	42
Setting up connectivity for a Microsoft SQL Server data source (Windows)	42
Setting up connectivity for a Microsoft Access source or a Microsoft Excel data source (Windows).	43
Setting up connectivity for an IMS, or a VSAM data source (Windows)	43
Managing the warehouse agent for non-DB2 sources	44
Verifying the warehouse agent (AIX, Solaris Operating Environment, Linux).	44
Switching between versions of the warehouse agent	44
Setting up connectivity for non-DB2 sources (AIX)	45
Setting up connectivity for a Sybase Adaptive Server source – without client (AIX)	45
Setting up connectivity for an Oracle source – without client (AIX)	45
Setting up connectivity for an Informix 9.2 source – without client (AIX)	46
Setting up connectivity for a Microsoft SQL Server source – without client (AIX)	47
Setting up connectivity for non-DB2 sources (Solaris Operating Environment, Linux).	47
Setting up connectivity for a Microsoft SQL Server source (Solaris Operating Environment, Linux)	47
Configuring non-DB2 warehouse sources	48
Configuring the ODBC driver for non-DB2 warehouse sources– without client (Windows).	48
Handling errors for Informix 9.2 – without client	49
Creating a target warehouse database (Windows)	49
Defining a warehouse that uses the Microsoft Access and warehouse target databases (Windows).	50
Importing table definitions from a Microsoft Access database (Windows)	51
Cataloging a Microsoft Access source database in ODBC (Windows)	51
Cataloging a target warehouse database for use with a Microsoft Access source database (Windows).	52

Using a Microsoft Excel spreadsheet as a warehouse source (Windows)	53
Cataloging a Microsoft Excel data source in ODBC (Windows)	53
Creating named tables for Microsoft Excel data sources (Windows)	54
Creating a target warehouse database for use with a Microsoft Excel data source (Windows).	54
Cataloging a target warehouse database for use with a Microsoft Excel data source (Windows).	55
Defining sources and targets to the Data Warehouse Center that use a Microsoft Excel data source (Windows)	55
Configuring the Red Brick client to access a Red Brick Warehouse database (Windows)	56
Configuring non-DB2 warehouse sources (AIX, Linux, Solaris Operating Environment)	57
Configuring the Red Brick client to access a Red Brick Warehouse database (AIX, Linux, Solaris Operating Environment)	57
Sample .odbc.ini file for the Red Brick ODBC driver	58
Defining a non-DB2 warehouse source in the Data Warehouse Center	58
Specifying database information for a non-DB2 warehouse source in the Data Warehouse Center	59
Defining warehouse sources for use with a DB2 federated system	60
Server definitions and nickname tables for warehouse sources accessed through a DB2 federated server	60
Defining source tables for federated system warehouse sources	61

Chapter 5. Setting up warehouse file sources 63

Warehouse agent support for file sources	63
Defining a file source to the Data Warehouse Center	63
Accessing remote data files	64
Accessing data files with the Copy file using FTP warehouse program	65
Setting up connectivity for file sources (Windows NT, Windows 2000, Windows XP)	65
Setting up connectivity for a z/OS or VM file source (Windows NT, Windows 2000, Windows XP)	65
Setting up connectivity for a local file source (Windows NT, Windows 2000, Windows XP)	66
Requirements for accessing a remote file from a file server (Windows)	66
Setting up connectivity for a remote file source (Windows NT, Windows 2000, Windows XP)	67
Setting up connectivity for file sources (AIX)	67
Setting up connectivity to a z/OS or VM file source (AIX)	67
Setting up connectivity for a local file source (AIX)	68
Setting up connectivity for a remote file source (AIX)	68
Setting up connectivity for file sources (Solaris Operating Environment, Linux).	69

Setting up connectivity for a z/OS or VM file source (Solaris Operating Environment, Linux)	69
Setting up connectivity for a local file source (Solaris Operating Environment, Linux)	69
Setting up connectivity for a remote file source (Solaris Operating Environment, Linux)	70
Example .odbc.ini file entry for a warehouse file source (AIX, Solaris Operating Environment, Linux)	70
Network File System protocol	71

Chapter 6. Setting up access to a warehouse 73

Setting up a DB2 Universal Database warehouse	73
Defining privileges for DB2 Universal Database warehouses	73
Connecting to DB2 Universal Database and DB2 Enterprise Server Edition warehouses	74
Setting up a DB2 for iSeries warehouse	74
Defining privileges to DB2 for iSeries warehouses	74
Setting up a DB2 Connect gateway site (iSeries)	75
Connecting to a DB2 for iSeries warehouse with DB2 Connect	75
Setting up a DB2 for z/OS warehouse	75
Setting up a DB2 Universal Database for z/OS warehouse	75
Defining privileges for a DB2 for z/OS warehouse	76
Connecting to a DB2 for z/OS warehouse database	76
Defining a DB2 for z/OS warehouse to the Data Warehouse Center	76
Setting up a warehouse on a database with multiple partitions	77
Defining privileges to a database with multiple partitions	77
Defining a database with multiple partitions to the Data Warehouse Center	78
Creating the Data Warehouse Center status table	78
Defining a warehouse for use with a federated system	79
Creating target tables in a remote database	80
Moving a target table from a federated database to a remote database	81
Updating a table in a remote database	82

Chapter 7. Setting up a warehouse target 83

Warehouse targets	83
Ordinary identifier support in the Data Warehouse Center	83
Delimited identifier support in the Data Warehouse Center	83
Edition columns	83
Supported warehouse targets	84
Defining a warehouse target	84
Warehouse primary and foreign keys	85
Running DB2 Warehouse Manager on a 64-bit DB2 instance	86

Chapter 8. Defining and running processes 87

Warehouse processes	87
Defining the transformation and movement of data within the Data Warehouse Center	88
Warehouse steps	88
Warehouse file programs	91
DB2 warehouse utilities	91
DB2 OLAP Server programs	92
Replication programs	93
Warehouse deprecated programs	94
Warehouse transformers	96
Statistical transformers	96
User-defined programs	97
Red Brick Warehouse utility programs	98
Column mapping	98
Running warehouse steps	100
Step promotion and testing	100
Verifying the results of a step that is run in test mode	101
Running steps that use transient tables as sources	101
Running a step from outside the Data Warehouse Center using the external trigger program	102
Syntax for starting the external trigger server	103
Syntax for starting the external trigger client	103
Data Warehouse Center messages for the external trigger program	105
Printing step information to a text file	106
Promoting warehouse processes	106
Managing step task flow	106
Step task flow	106
Cascading processes	107
Process task flow	107
Monitoring the progress of steps and processes	108
Monitoring the progress of a step or process using the Work in Progress window	108
Step and process error messages	109
Sampling data	109

Chapter 9. Selecting, inserting, and updating source data in a target table . 111

SQL steps	111
Defining an SQL step	111
Incremental commit	112
Joining source columns	112
Defining a join	112
Removing a join	113
Transforming codes	114
Join types	114
Star joins	116
Filtering data	117
Specifying search conditions when filtering data	118
Creating composite summary steps	118
Defining a calculated column and adding it to a step	119
Adding a function to an expression in the Expression Builder	120
Adding a constant to an expression	120

Chapter 10. Loading and exporting data	121
Data Warehouse Center load and export programs	121
Exporting data	122
Defining values for a DB2 UDB export utility	122
Defining values for the Data export with ODBC to file utility	122
Defining values for the Red Brick SQL Export utility	123
Loading data	124
Defining values for a DB2 Universal Database load utility	124
Defining a DB2 for iSeries Data Load Insert utility	125
Defining a DB2 for iSeries Data Load Replace utility	126
Modstring parameters for DB2 for iSeries load utilities	127
Trace files for the DB2 for iSeries load utilities	128
Viewing trace files for the DB2 for iSeries load utilities	129
Defining a file extension to Client Access/400	129
Defining a DB2 for z/OS load utility	130
Copying data between DB2 utilities	131
Defining values for the Red Brick TMU Load utility	132
Chapter 11. Moving files and tables	133
Manipulating files using FTP or the Submit JCL jobstream warehouse program	133
Accessing a remote file on a secure UNIX or UNIX System Services system	134
Replication	134
Replication in the Data Warehouse Center	135
Replication source tables in the Data Warehouse Center	136
Setting up replication in the Data Warehouse Center	137
Creating replication control tables	138
Defining a replication step in the Data Warehouse Center	139
Using a replication step in a process	139
Replication password files	140
Chapter 12. Transforming data	143
Using transformers in the Data Warehouse Center	143
Transforming target tables	143
Clean Data transformer	143
Rules tables for Clean Data transformers	148
Key columns	149
Period table	149
Invert Data transformer	150
Pivot data transformer	151
FormatDate transformer	152
Changing the format of a date field	152
Specifying the input format for a date field	153
Specifying the output format for a date field	153
Cleaning name and address data	153
Cleaning name and address data with the Trillium Software System	153

Trillium Software System components	155
Trillium metadata	156
Importing Trillium metadata	156
Trillium Batch System JCL files	157
Example: Job step that includes a SYSTERM DD statement	158
Trillium Batch System user-defined program Parameters for the Trillium Batch System script or JCL	159
Error handling for Trillium Batch System programs	160
Error return codes for name and address cleansing	161

Chapter 13. Calculating statistics . . . 163

Defining statistical transformers in the Data Warehouse Center	163
ANOVA transformer	163
Calculate Statistics transformer	164
Calculate Subtotals transformer	164
Chi-square transformer	166
Correlation analysis	166
Moving averages	168
Regression transformer	169

Chapter 14. Exporting and importing metadata in the Data Warehouse Center . . . 171

Metadata export capabilities	171
Tag language and XML files	172
Importing metadata	173
Publishing warehouse metadata	174
Increasing the application heapsize for publishing Data Warehouse Center metadata	175
MQSeries	175
MQSeries data	175
Creating views for MQSeries messages	176
Importing MQSeries messages and XML metadata	176
MQSeries stored procedure	178
Error return codes for the MQXMLXF stored procedure	178
Error logging for MQSeries	179
Importing ERwin metadata	180
IBM ERwin 3.5.2 metadata extract program	180
Running the ERwin metadata import commands db2erwinimport command for ERwin 4.0 or later	181
Creating tag language files for the IBM ERwin Metadata Extract program	184
Merging ERwin metadata with existing database data	185
Returning the ER1 file to a writable state	185
iw2imp2 command syntax for ERwin 3.5.2 to import metadata into the Data Warehouse Center	185
Changing the tag language file to change the DB2 database definition to a source in the Data Warehouse Center	186
IBM ERwin Metadata Extract program errors	186

Chapter 15. Extending the Data Warehouse Center	189
User-defined programs	189
What is a program group?	189
What is a user-defined program?	189
Using a user-defined program in the Data Warehouse Center	190
Agent sites for user-defined programs	190
Parameters for user-defined programs	191
Writing your own program for use with the Data Warehouse Center	191
Microsoft OLE DB and Data Transaction Services support	197
Creating views for OLE DB table functions	198
Creating views for DTS packages	198

Chapter 16. Creating a star schema from within the Data Warehouse Center	201
Star schemas	201
Designing the warehouse schema in the Data Warehouse Center	202
Warehouse schemas	202
Adding tables and views to the warehouse schema	202
Exporting a warehouse schema to the DB2 OLAP Integration Server	203
Working with a warehouse schema in the DB2 OLAP Integration Server	204
Completing the design for your cube	204
Creating an outline in the DB2 OLAP Integration Server	204
Creating a load script to load data into the cube in the DB2 OLAP Integration Server	205
Creating a batch file to load the command script for the DB2 OLAP Integration Server	206
Exporting the metadata to the Data Warehouse Center	207
Loading the multidimensional cube from the Data Warehouse Center	208
Data Warehouse Center objects created by the DB2 OLAP Integration Server export process	208
Creating a schedule to populate a cube	209
About publishing metadata about your warehouse schema	209

Chapter 17. Maintaining the warehouse database	211
Reorganizing data	211
Defining values for DB2 UDB REORG or DB2 for z/OS REORG utilities	211
Defining values for a DB2 for z/OS Utility	211
Parameters for a DB2 for z/OS Utility	211
Updating system catalog statistics	215
Defining values for a DB2 UDB RUNSTATS utility	215
Defining values for a DB2 for z/OS RUNSTATS utility	215

Chapter 18. Managing the control database	217
Backing up data	217
Stopping the Data Warehouse Center services (Windows)	217
Backing up the warehouse control database	217
Expanding your warehouse	217
Adding or deleting administrative interfaces and warehouse agents in the Data Warehouse Center	218
Initializing a warehouse database	218
Changing the active warehouse control database	218
Initializing a warehouse control database during installation	219
Migrating a DB2 Universal Database Version 7 Control Database to DB2 Universal Database Version 8 when Version 7 and Version 8 are installed on different systems	220
Data Warehouse Center configuration	221
Data Warehouse Center configuration	221

Chapter 19. Refreshing an OLAP Server database	223
Loading data into the OLAP server database from the Data Warehouse Center	223
Running calculations on OLAP server data from the Data Warehouse Center	224
Loading data from a flat file into an OLAP Server database	224
Loading data from a table into an OLAP server database	225
Updating an OLAP server outline from the Data Warehouse Center	225

Chapter 20. Data Warehouse Center logging and trace data	227
The basic logging function	227
Warehouse log files	227
Viewing run-time errors using the basic logging function	227
Viewing build-time errors using the basic logging function	228
Viewing log entries in the Data Warehouse Center	228
Data Warehouse Center component trace data	228
Component trace data	228
Running warehouse agents as a user process (Windows)	230
Running a Data Warehouse Center component trace	230
Error logging for warehouse programs and transformers	231
Tracing Apply program errors	232
Start error trace files	232

Appendix A. Metadata mappings	233
Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center	233

Mapping ERwin Version 4.0 and later object attributes to Data Warehouse Center tags	234
Mapping ERwin Version 3.5.2 object attributes to Data Warehouse Center tags	236
Trillium DDL to Data Warehouse Center metadata mapping	237
Metadata mappings between the Data Warehouse Center and CWM XML objects and properties	238

Appendix B. Defining values for warehouse deprecated programs and transformers. 241

Defining values for Visual Warehouse 5.2 compatible programs	241
Defining values for a Visual Warehouse 5.2 DB2 UDB Data Export (VWPEXPT1) program	241
Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Insert (VWPLOADI) program	242
Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Replace (VWPLOADR) program	244
File type modifiers	245
Defining values for a Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program	246
Defining values for a Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) program	247

Defining values for a Visual Warehouse 5.2 Load flat file into DB2 Enterprise Server Edition (VWPLDPR) program (AIX)	247
Defining values for the Data Warehouse Center Version 7.2 Clean Data transformer	249
Data Warehouse Center Version 7.2 Clean Data transformer	249
Rules tables for Data Warehouse Center Version 7.2 Clean Data transformer	250
Defining a Data Warehouse Center Version 7.2 Clean Data transformer	252

Appendix C. Data Warehouse Center environment structure 253

Data Warehouse Center environment variables	253
Data Warehouse Center registry updates	253
Data Warehouse Center configuration files	254

Notices 255

Trademarks	257
----------------------	-----

Index 259

Contacting IBM 267

Product information	267
-------------------------------	-----

About this book

This book describes the steps that are required to use the IBM® DB2 Universal Database™ Data Warehouse Center to build and maintain a warehouse. A *warehouse* is a database that contains informational data that is extracted and transformed from your operational data sources.

In this book, the term *z/OS* refers to both *z/OS*® and *OS/390*®.

Who should read this book

This book is intended for the administrators and warehouse designers who use the Data Warehouse Center to build a warehouse. Readers should have a basic understanding of relational database management systems, some knowledge of SQL, and access to the experts who can establish connectivity to various data sources.

To use the statistical transformers, you should be familiar with statistical techniques.

Prerequisite publications

Before you read this book, read *DB2 Universal Database Quick Beginnings* to install the Data Warehouse Center. If you have the DB2 Warehouse Manager, read the *DB2 Warehouse Manager Installation Guide* to install agents and transformers. If you use replication, read the *IBM DB2 Replication Guide and Reference*.

In addition to this book, you will need documentation for:

- The Microsoft® Windows NT®, Windows® 2000, or Windows XP operating system.
- If you are running the server on AIX®, you will need the documentation for the AIX operating system.
- The databases you plan to use as sources and targets.
- The connectivity software you plan to use.

Chapter 1. About data warehousing

IBM® DB2 Universal Database™ offers the Data Warehouse Center, a component that automates data warehouse processing. You can use the Data Warehouse Center to define the processes that move and transform data for the warehouse. Then, you can use the Data Warehouse Center to schedule, maintain, and monitor these processes.

This chapter provides an overview of data warehousing and data warehousing tasks.

What solutions does data warehousing provide?

The systems that contain *operational data* (the data that runs the daily transactions of your business) contain information that is useful to business analysts. For example, analysts can use information about which products were sold in which regions at which time of year to look for anomalies or to project future sales.

However, several problems can arise when analysts access the operational data directly:

- Analysts might not have the expertise to query the operational database. For example, querying IMS™ databases requires an application program that uses a specialized type of data manipulation language. In general, the programmers who have the expertise to query the operational database have a full-time job in maintaining the database and its applications.
- Performance is critical for many operational databases, such as databases for a bank. The system cannot handle users making ad hoc queries.
- The operational data generally is not in the best format for use by business analysts. For example, sales data that is summarized by product, region, and season is much more useful to analysts than the raw data.

Data warehousing solves these problems. In *data warehousing*, you create stores of *informational data*. Informational data is data that is extracted from the operational data and then transformed for decision making. For example, a data warehousing tool might copy all the sales data from the operational database, clean the data, perform calculations to summarize the data, and write the summarized data to a target in a separate database from the operational data. Users can query the separate database (the *warehouse*) without impacting the operational databases.

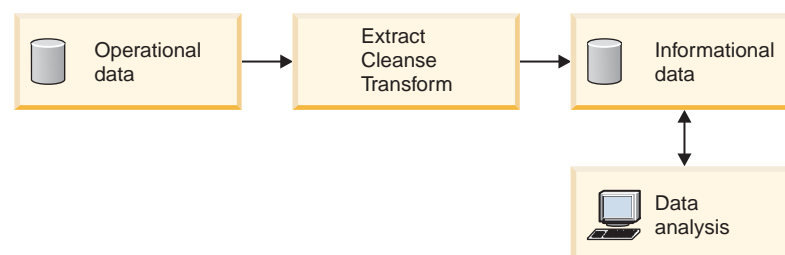


Figure 1. The path from operational data to data analysis

Related concepts:

About data warehousing

- “Data warehouse objects” on page 2

Warehouse tasks

Creating a data warehouse involves the following tasks:

- Identifying the source data (or operational data) and defining it for use as warehouse sources.
- Creating a database to use as the warehouse and defining warehouse targets.
- Defining a subject area for groups of processes that you will define in your warehouse.
- Specifying how to move and transform the source data into its format for the warehouse database by defining steps in the processes.
- Testing the steps that you define and scheduling them to run automatically.
- Administering the warehouse by defining security and monitoring database usage using the Work in Progress notebook.

| If you have DB2[®] Warehouse Manager, you can create an information catalog of
| the data in the warehouse. An information catalog is a database that contains
| business metadata. Business metadata helps users identify and locate data and
| information available to them in the organization. Data Warehouse Metadata can
| be published to the information catalog. The information catalog can be searched
| to determine what data is available in the warehouse.

| You can also define a star schema model for the data in the warehouse. A star
| schema is a specialized design that consists of multiple dimension tables, which
| describe aspects of a business, and one fact table, which contains the facts or
| measurements about the business. For example, for a manufacturing company,
| some dimension tables are products, markets, and time. The fact table contains
| transaction information about the products that are ordered in each region by
| season.

Related concepts:

- “Warehouse steps” on page 88

Related tasks:

- “Steps and tasks: Data Warehouse Center help”

Data warehouse objects

The following sections describe the objects that you will use to create and maintain your data warehouse.

Subject areas

A subject area identifies and groups the processes that relate to a logical area of the business. For example, if you are building a warehouse of marketing and sales data, you define a Sales subject area and a Marketing subject area. You then add the processes that relate to sales under the Sales subject area. Similarly, you add the definitions that relate to the marketing data under the Marketing subject area.

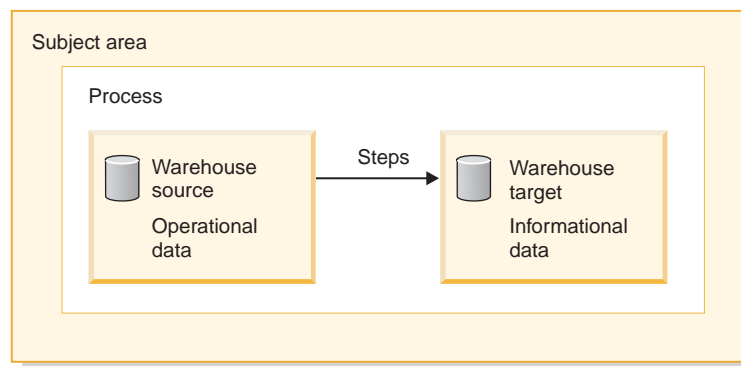


Figure 2. The hierarchy of subject areas

Warehouse sources

Warehouse sources identify the tables and files that will provide data to your warehouse. The Data Warehouse Center uses the specifications in the warehouse sources to access the data. The sources can be almost any relational or nonrelational source (table, view, or file), or WebSphere® Site Analyzer source that has connectivity to your network.

Warehouse targets

Warehouse targets are database tables or files that contain data that has been transformed. You can use warehouse targets to provide data to other warehouse targets. For example, a central warehouse can provide data to departmental servers, or a main fact table in the warehouse can provide data to summary tables.

Warehouse control databases

The warehouse control database contains the control tables that are required to store the Data Warehouse Center metadata. Starting in the Data Warehouse Center Version 8.2, the warehouse control database must be a UTF-8 (Unicode Transformation Format, or Unicode) database. This requirement provides expanded language support for the Data Warehouse Center. If you try to log on to the Data Warehouse Center using a database that is not in Unicode format, you will receive an error message that you cannot log on. You can use the Warehouse Control Database Management tool to migrate the metadata from a specified database into a new Unicode database.

Warehouse agents and agent sites

Warehouse agents manage the flow of data between the data sources and the target warehouses. Warehouse agents are available on the AIX®, Linux, iSeries™, z/OS™, Windows® NT, Windows 2000, and Windows XP operating systems, and for the Solaris™ Operating Environment. The agents use Open Database Connectivity (ODBC) drivers or DB2® CLI to communicate with different databases.

Several agents can handle the transfer of data between sources and target warehouses. The number of agents that you use depends on your existing connectivity configuration and the volume of data that you plan to move to your warehouse. Additional instances of an agent can be generated if multiple processes that require the same agent are running simultaneously.

About data warehousing

Agents can be local or remote. A local warehouse agent is an agent that is installed on the same workstation as the warehouse server. A remote warehouse agent is an agent that is installed on another workstation that has connectivity to the warehouse server.

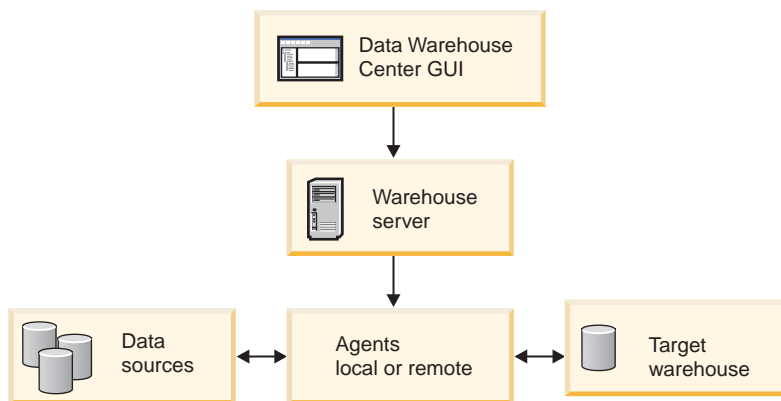


Figure 3. The relationship between agents, data sources, target warehouses, and the warehouse server

An agent site is a logical name for a workstation where agent software is installed. The agent site name is not the same as the TCP/IP host name. A single workstation can have only one TCP/IP host name. However, you can define multiple agent sites on a single workstation. A logical name identifies each agent site.

The default agent site, named the Default DWC Agent Site, is a local agent that the Data Warehouse Center defines during initialization of the warehouse control database.

Processes and steps

A process contains a series of steps that performs a transformation and movement of data for a specific warehouse use. In general, a process moves source data into the warehouse. Then, the data is aggregated and summarized for warehouse use. A process can produce a single flat table or a set of summary tables. A process might also perform some specific type of data transformation.

A *step* is the definition of a single operation within the warehouse. By using SQL statements or calling programs, steps define how you move data and transform data. When you run a step, a transfer of data between the warehouse source and the warehouse target, or any transformation of that data, can take place.

A step is a logical entity in the Data Warehouse Center that defines:

- A link to its source data.
- The definition of and a link to the output table or file.
- The mechanism (either an SQL statement or a program) and definition for populating the output table or file.
- The processing options and schedule by which the output table or file is populated.

Suppose that you want Data Warehouse Center to perform the following tasks:

1. Extract data from different databases.
2. Convert the data to a single format.

3. Write the data to a table in a data warehouse.

You would create a process that contains several steps. Each step performs a separate task, such as extracting the data from a database or converting it to the correct format. You might need to create several steps to completely transform and format the data and put it into its final table.

When a step or a process runs, it can affect the target in the following ways:

- Replace all the data in the warehouse target with new data
- Append the new data to the existing data
- Append a separate edition of data
- Update existing data

You can run a step on demand, or you can schedule a step to run:

- At a set time
- Only one time
- Repeatedly, such as every Friday
- In Sequence, so that when one step finishes running, the next step begins running
- Upon completion, either successful or not successful, of another step

If you schedule a process, the first step in the process runs at the scheduled time.

The following sections describe the various types of steps that you will find in the Data Warehouse Center.

SQL steps

The Data Warehouse Center provides two types of SQL steps. The SQL Select and Insert step uses an SQL SELECT statement to extract data from a warehouse source and generates an INSERT statement to insert the data into the warehouse target table. The SQL Select and Update step uses an SQL SELECT statement to extract data from a warehouse source and update existing data in the warehouse target table.

Program steps

The Data Warehouse Center provides several types of program steps: DB2 for iSeries programs, DB2 for z/OS programs, DB2 Universal Database™ programs, Visual Warehouse™ 5.2 DB2 programs, OLAP Server programs, File programs, and Replication programs. These steps run predefined programs and utilities. The warehouse programs for a particular operating system are packaged with the agent for that operating system. You install the warehouse programs when you install the agent code.

Transformer steps

Transformer steps are stored procedures and user-defined functions that specify statistical or warehouse transformers that you can use to transform data. You can use transformers to clean, invert, and pivot data; generate primary keys and period tables; and calculate various statistics.

In a transformer step, you specify one of the statistical or warehouse transformers. When you run the process, the transformer step writes data to one or more warehouse targets.

About data warehousing

User-defined program steps

A user-defined program step is a logical entity within the Data Warehouse Center that represents a business-specific transformation that you want the Data Warehouse Center to start. Because every business has unique data transformation requirements, businesses can choose to write their own program steps or to use tools such as those provided by other companies, such as ETI or Vality.

For example, you can write a user-defined program that will perform the following functions:

1. Export data from a table.
2. Manipulate that data.
3. Write the data to a temporary output resource or a warehouse target.

Related concepts:

- “What solutions does data warehousing provide?” on page 1
- “Warehouse tasks” on page 2
- “What is a user-defined program?” on page 189
- “What is a program group?” on page 189

Chapter 2. Setting up your warehouse

The process of building a warehouse is iterative. You might begin with a certain set of assumptions and build a warehouse that reflects those assumptions. Eventually, you might want to change these assumptions and, therefore, change the warehouse. Because the Data Warehouse Center is easy to install and use, you can make changes to the warehouse design without having to rebuild your warehouse. You can iteratively evaluate your assumptions and change your warehouse as many times as you need to. The Data Warehouse Center adjusts with your needs.

To build a warehouse:

1. Start the Data Warehouse Center.
2. Define the agent sites.
3. Define Data Warehouse Center security.
4. Define subject areas.

Each of these steps is explained later.

Starting the Data Warehouse Center

To start the Data Warehouse Center, you need to start its components. Some components start automatically, and some start manually.

After the warehouse server and logger are installed, they start automatically when you start Windows NT, Windows 2000, or Windows XP. The warehouse agent can start automatically or manually. You open the Data Warehouse Center administrative interface manually from the DB2 Control Center.

Prerequisites:

To start the Data Warehouse Center, you must start the warehouse server and logger if they are not started automatically.

You must also start the warehouse daemon, if you are using an iSeries or z/OS warehouse agent, or if you are using a remote Windows NT or Windows 2000 warehouse agent that must be started manually.

Procedure:

To start the Data Warehouse Center:

1. Start the Data Warehouse Center administrative interface.
2. Define an agent site to the Data Warehouse Center.
3. Define security for the Data Warehouse Center.

Related concepts:

- “Data Warehouse Center security” on page 20

Related tasks:

- “Starting and stopping the warehouse server and logger (Windows)” on page 9
- “Defining an agent site” on page 18

Setting up your warehouse

- “Defining warehouse security” on page 22
- “Starting and stopping the warehouse server and logger (AIX)” on page 9
- “Starting the Data Warehouse Center administrative interface” on page 8

Starting the Data Warehouse Center administrative interface

You must start the Data Warehouse Center administrative interface before you can work with warehouse objects. On Windows, you can start the Data Warehouse Center administrative interface from the GUI. On AIX or the Solaris Operating Environment, you start the Data Warehouse Center administrative interface from the command line.

Prerequisites:

Verify that the warehouse server and logger are running before you start the Data Warehouse Center administrative interface.

Procedure:

To start the Data Warehouse Center administrative interface:

1. Start the Data Warehouse Center:
 - On Windows, click **Start** → **Programs** → **IBM DB2** → **Business Intelligence Tools** → **Data Warehouse Center**.
 - On AIX or the Solaris Operating Environment, enter the following command:
db2dwc

The Data Warehouse Center Logon window opens.

2. If you are logging on for the first time, click **Advanced** to open the Advanced window.
3. In the **Control database** field, type the name of the warehouse control database. If the warehouse control database is a remote database, then you must catalog the node and the control database. If the warehouse control database is a local database, then it requires no additional configuration.
4. In the **Server host name** field, type the TCP/IP host name for the workstation where the warehouse server is installed.
5. Click **OK** to save your changes and close the Advanced window.

The next time that you log on, the Data Warehouse Center will use the settings that you specified in the Advanced window.
6. In the **User ID** field of the Logon window, type a valid Data Warehouse Center user ID.

If you are logging on for the first time, type the default warehouse user ID.
7. In the **Password** field, type the password for the user ID.
8. Click **OK** to save your changes and close the Data Warehouse Center Logon window.

If you are using the Data Warehouse Center or the Information Catalog Center on an AIX operating system, you might receive the error SQL1224N. If you receive this error, refer to the message text for a list of possible causes. If the problem involves the number of shared memory segments, see the Administrative API reference documentation for information regarding threaded applications with concurrent access. This documentation provides solutions for problems involving

the number of shared memory segments, including setting the EXTSHM environment variable or defining a loopback to the database being accessed.

Related tasks:

- “Opening the Data Warehouse Center: Data Warehouse Center help”

Related reference:

- “Threaded Applications with Concurrent Access” in the *Administrative API Reference*

Starting and stopping the Data Warehouse Center server and logger

This section describes how to start and stop the warehouse server and logger on Windows and AIX operating systems. It also describes how to verify that the AIX warehouse server and logger daemons are running.

Starting and stopping the warehouse server and logger (Windows)

The warehouse server and the warehouse logger run as services on Windows. To start the services, you must restart the system after you initialize the warehouse control database. Then the warehouse server and logger will start automatically every time you start Windows unless you change them to a manual service, or unless you stop the DB2 server. If you stop the DB2 server, connections to local and remote databases are lost. To restore connections you must stop and restart the warehouse server and logger after you stop and restart the DB2 server.

Procedure:

To manually start the warehouse server and logger on Windows:

1. From the Windows desktop, click **Start** —> **Settings** —> **Control Panel** —> **Services**.
2. Click **DB2 Warehouse Server** in the Services window.
3. Click **Start**, and click **OK** to start both the warehouse server and the warehouse logger.

To manually stop the warehouse server and the warehouse logger, repeat steps 1 and 2, and then click **Stop**.

Starting and stopping the warehouse server and logger (AIX)

On AIX, you can manually start or stop the warehouse server (iwh2serv) and logger (iwh2log) daemons using the db2vwsvr shell script. When starting the daemons, the db2vwsvr start command adds an entry in the inittab file with identifier db2vwsvr. As a result, the inittab daemon will rerun the db2vwsvr start command automatically if the warehouse server daemon stops.

The db2vwsvr shell script first runs the IWH.environment script to initialize the environment. It assumes that these scripts are in the same directory. The db2vwsvr and IWH.environment script files are installed in the bin subdirectory of the DB2 installation directory. If the IWH.environment script file is changed, you must restart the daemons by running the db2vwsvr start command.

Procedure:

Setting up your warehouse

To start the warehouse server and logger daemons, enter the following command at the AIX command prompt:

```
db2vwsvr start
```

To stop the warehouse server and logger daemons, enter the following command at the AIX command prompt:

```
db2vwsvr stop
```

When you stop the daemons by running the `db2vwsvr stop` command, the `iwh2serv` and `iwh2log` processes are restarted, and the entry in the `inittab` file with identifier `db2vwsvr` is updated. Also, the `inittab` entry with identifier `db2vwsvr` is removed. As a result, the warehouse server and logger daemons will not be restarted automatically. You need to run the `db2vwsvr start` command again to reinitialize the `inittab` entry with the `db2vwsvr` identifier.

Verifying that the warehouse server and logger daemons are running (AIX)

You can verify whether the AIX warehouse server and warehouse logger daemons are running.

Prerequisites:

You must be logged on as the DB2 instance owner.

Procedure:

To verify that the warehouse server and logger daemons are running:

1. To display the server and logger processes, enter the following command at the AIX command prompt:

```
db2 list applications
```

If the warehouse server and warehouse logger daemons are running, the command displays the processes `iwh2serv` (server) and `iwh2log` (logger):

Auth Id	Application Name	Appl. Handle	Application Id	DB Name	# of Agents
DB2IV81	iwh2serv	4	*LOCAL.db2iv81.0A0871221727	DWCTRLDB	1
DB2IV81	iwh2log	3	*LOCAL.db2iv81.07B631221725	DWCTRLDB	1

2. To display the `vwkernel` process, enter the following command at the AIX command prompt:

```
$ netstat -a|grep vwkernel
```

If the warehouse server and warehouse logger are running, the command displays the `vwkernel` process in the listen state:

```
tcp4      0      0 *.vwkernel      *.*              LISTEN
```

3. To display the `vwlogger` process, enter the following command at the AIX command prompt:

```
$ netstat -a|grep vwlogger
```

If the warehouse server and warehouse logger are running, the command displays the `vwlogger` process in the listen state:

```
tcp4      0      0 *.vwlogger      *.*              LISTEN
```

If the daemons are not started, you can also check the DB2VWSVR.LOG, IWH2LOG, and IWH2SERV.LOG files in the logging directory for any errors. The logging directory is specified by the environment variable VWS_LOGGING. The default value is /var/IWH.

Starting the warehouse agent daemon

This section describes how to start the warehouse agent daemon.

Warehouse agent daemons

The warehouse server comes with a warehouse agent, called the *local agent*. The local agent is defined as the default warehouse agent for all Data Warehouse Center activities. The local agent starts automatically when the warehouse server starts.

If you installed an AIX[®], Linux, or the Solaris[™] Operating Environment warehouse agent, the warehouse agent daemon starts automatically.

If you are using a remote Windows[®] NT, Windows 2000, or Windows XP warehouse agent, you can select whether to start the warehouse agent daemon automatically or manually during installation.

The Data Warehouse Center does not provide a way to start the iSeries[™] or zSeries[®] warehouse agent automatically.

Related concepts:

- “Agent site configurations” on page 17
- “Agent sites for user-defined programs” on page 190

Related tasks:

- “Defining an agent site” on page 18

Connectivity requirements for the warehouse server and the warehouse agent

The warehouse server uses TCP/IP to communicate with the warehouse agent and the warehouse agent daemon. For this communication to occur, the warehouse server must be able to recognize the fully qualified host name of the warehouse agent. Also, the warehouse agent must be able to recognize the fully qualified host name of the warehouse server.

By default, the warehouse server sends messages to the warehouse agent daemon on port 11001 and receives responses on port 11000. The warehouse logger uses port 11002. If another application uses one of the default Data Warehouse Center port numbers, you can change the port number that the Data Warehouse Center uses.

Related concepts:

- “Warehouse agent daemons” on page 11

Starting the warehouse agent daemon (Windows)

If you installed a Windows warehouse agent other than the local agent, you can manually start the warehouse agent daemon just as you start any Windows service.

Prerequisites:

You must install the warehouse agent, and the agent must be set to start manually. Otherwise, the agent starts automatically when you start Windows.

Procedure:

To manually start the Windows warehouse agent daemon:

1. Click **Start** —> **Settings** —> **Control Panel** —> **Services**.
2. Scroll down the list until you find **DB2 Warehouse Agent**. Make any changes you want, and click **OK**.

Starting the iSeries warehouse agent daemon

After you install the iSeries warehouse agent, you must start the warehouse agent daemon using the STRVWD command. The STRVWD command starts QIWH/IWHVWD (the warehouse agent daemon) in the QIWH subsystem as a background job. This causes all warehouse agent processes that are started by the warehouse agent daemon to start in the QIWH subsystem.

Prerequisites:

The user profile that starts the warehouse agent daemon must have *JOBCTL authority.

Procedure:

To start the warehouse daemon, enter STRVWD at an iSeries command prompt.

Verifying that the iSeries warehouse agent started

After you enter the command to start the iSeries warehouse agent, you can verify that it started successfully.

Procedure:

To verify that the iSeries warehouse agent is running, enter WRKACTJOB at an iSeries command prompt, and look for the jobs VWD and IWH4MSGQ. If these jobs are displayed, the warehouse agent daemon is started.

Verifying that the iSeries warehouse agent daemon is still running

Occasionally, you might want to verify that the iSeries warehouse agent daemon that you started is still running. For example, you use the iSeries warehouse agent throughout your work week, then go home for the weekend. When you return the following Monday, you should verify that the warehouse agent daemon is active before you initiate a new agent process.

Procedure:

To verify that the iSeries warehouse agent daemon is active:

1. At an iSeries command prompt, enter WRKACTJOB. The active jobs are displayed.
2. Look for the function PGM-IWHVWD that is associated with the user ID you used when you started the warehouse agent daemon. If the function is displayed, the warehouse agent is active.

Configuring TCP/IP on z/OS

You must configure TCP/IP on z/OS so that the warehouse server can communicate with the warehouse agent and the warehouse agent daemon.

Procedure:

To configure TCP/IP on z/OS:

1. Open the `/etc/services` file, or the `TCPIP.ETC.SERVICES` file in the z/OS environment.
2. Add the following services to the file:

Table 1. TCP/IP services to add

Port name	Port number
vwkernel	11000/tcp
vwd	11001/tcp
vwlogger	11002/tcp

Related concepts:

- “Warehouse agent daemons” on page 11
- “Connectivity requirements for the warehouse server and the warehouse agent” on page 11

Related tasks:

- “Starting the Data Warehouse Center” on page 7
- “Starting the zSeries agent daemon in the foreground” on page 13
- “Starting the zSeries warehouse agent daemon in the background” on page 14
- “Verifying that the zSeries warehouse agent is running” on page 14
- “Verifying communication between the warehouse server and the warehouse agent” on page 15

Starting the zSeries agent daemon in the foreground

After you configure your system for the zSeries warehouse agent, you can start the warehouse agent daemon in the foreground or in the background. This task describes how to start the zSeries warehouse agent daemon in the foreground.

Prerequisites:

- Your system must be configured for the zSeries warehouse agent.
- Both the zSeries warehouse agent and zSeries agent daemon run on the UNIX® System Services (USS) platform.

Setting up your warehouse

Procedure:

To start the warehouse agent daemon in the foreground:

1. Use Telnet to access USS on z/OS through the z/OS host name and USS port.
2. Enter vwd on the command line to start the agent daemon in the foreground.

Related tasks:

- “Starting the agent daemon as a z/OS started task” in the *DB2 Warehouse Manager Standard Edition Installation Guide*

Starting the zSeries warehouse agent daemon in the background

After you configure your system for the zSeries warehouse agent, you can start the warehouse agent daemon in the foreground or in the background. This task describes how to start the zSeries warehouse agent daemon in the background.

Prerequisites:

- Your system must be configured for the zSeries warehouse agent.
- Both the zSeries agent and zSeries agent daemon run on the UNIX[®] System Service (USS) platform.

Procedure:

To start the warehouse agent daemon in the background, enter `vwd>[/usr/lpp/DWC/logs/vwd.log] 2>&1 &` on the command line under a UNIX System Services shell.

`[/usr/lpp/DWC/logs/vwd.log]` is the directory and path name where you want to send daemon messages.

Verifying that the zSeries warehouse agent is running

Occasionally, you might want to verify that the zSeries warehouse agent daemon that you started is still running. For example, you use the zSeries warehouse agent throughout your work week, then go home for the weekend. When you return the following Monday, you should verify that the warehouse agent daemon is active before you initiate a new agent process.

You can verify that the zSeries warehouse agent is running from a UNIX shell, or from a console. This task describes how to verify that the warehouse agent is running from a UNIX shell.

Procedure:

To verify from a UNIX shell that the warehouse agent daemon is running, enter the following command on a UNIX shell command line:

```
ps -e | grep vwd
```

If the warehouse agent daemon is running, and you are authorized to see the task, a message similar to the following message will be returned:

```
$ ps -ef | grep vwd
MVSUSR2      198  16777537  - 13:13:22 ttyp0013  0:00 grep vwd
MVSUSR2      16777446 16777538  - 09:57:21 ttyp0002  0:00 vwd
```

If the warehouse agent daemon is not running, or if you are not authorized to see the task, a message similar to the following message will be returned:

```
$ ps -ef | grep vwd
MVSUSR2      198  16777537  - 13:13:22 ttyp0013  0:00 grep vwd
```

To verify from an z/OS console that the warehouse agent daemon is running, enter the following command at the z/OS command prompt:

```
D OMVS,A=ALL
```

If the warehouse agent daemon is running, a task with the string vwd is displayed in the message that is returned. A message similar to the following example is displayed:

```
D OMVS,A=ALL
BPX0040I 13.16.15 DISPLAY OMVS 156
OMVS      000E ACTIVE          OMVS=(00)
USER      JOBNAME  ASID      PID      PPID STATE   START   CT_SECS
MVSUSR2  MVSUSR24 00C5      16777446 16777538 HRI    09.57.20   .769
LATCHWAITPID=          0 CMD=vwd
```

Verifying communication between the warehouse server and the warehouse agent

The Data Warehouse Center uses TCP/IP to communicate with a remote agent site. For this communication to take place, the warehouse server must be able to recognize the fully qualified host name of the agent site. Also, the agent site must be able to recognize the fully qualified host name of the warehouse server.

Procedure:

To verify that one site recognizes the fully qualified host name of the other site, enter the **ping** command at a command prompt.

For example, the fully qualified host name for a warehouse agent site is `abc.xyz.commerce.com`. To verify that the warehouse server recognizes the fully qualified host name of the agent site, from a DOS command prompt, enter:
`ping abc.xyz.commerce.com`

Ensure that you verify communication from both the agent site to the warehouse server workstation and vice versa.

Stopping the warehouse agent daemon

This section describes how to stop the warehouse agent daemon.

Stopping the warehouse agent daemon (Windows)

You need to stop the warehouse agent daemon when you change the environment variables for the warehouse agent and warehouse agent daemon on Windows.

Procedure:

Setting up your warehouse

To stop the Windows warehouse agent daemon:

1. Click **Start** → **Settings** → **Control Panel** → **Services**.
2. Select **DB2 Warehouse Agent**.
3. Click **Stop**.
The warehouse agent daemon stops.
4. Click **OK**.

Stopping the warehouse agent daemon (AIX, Solaris Operating Environment, Linux)

The only time you must stop the AIX, Linux, or Solaris Operating Environment warehouse agent daemon is when you want to change the environment variables for the warehouse agents and their corresponding warehouse agent daemon.

Procedure:

To change the environment variables for one of the warehouse agents and its corresponding warehouse agent daemon:

1. Change the environment variables for both the warehouse agent and the warehouse agent daemon by editing the IWH.environment file.
2. Display the process ID for the warehouse agent daemon. At an AIX, Linux, or Solaris Operating Environment command line, enter:

```
ps -ef|grep vwd
```

The process ID is displayed.

3. Stop the warehouse agent daemon. At a command line enter:

```
kill processid
```

where *processid* is the process ID that was displayed in step 2.

When the warehouse agent daemon stops, it immediately restarts by default. When the warehouse agent daemon restarts, it uses the new environment variables.

Stopping the iSeries warehouse agent daemon

Occasionally, you might need to stop the iSeries warehouse agent daemon.

Procedure:

To stop the warehouse agent daemon, enter the following command at an iSeries command prompt.

```
ENDVWD
```

When you enter this command, either the warehouse agent daemon stops, or a list of jobs is displayed. If a list of jobs is displayed, end the job that has ACTIVE status.

Stopping the warehouse agent daemon (zSeries)

Occasionally, you might need to stop the zSeries warehouse agent daemon.

Procedure:

To stop the agent daemon:

1. Determine the daemon process ID by entering one of the following commands.

From a UNIX System Services session, enter the following command:
`ps -ef | grep vwd`

From the system console or the System Display and Search Facility (SDSF) log panel, enter the following command:
`D OMVS,A=ALL`

2. Stop the daemon process by entering the following command from the UNIX System Services session:

`kill [-9] pid` on the command line, where `pid` is the daemon process ID.

To stop the z/OS daemon when it is running in the foreground on a UNIX System Services shell, press `Ctrl+c` from the shell where the daemon is running.

Defining agent sites

When you set up your data warehouse, you must define the agent sites the warehouse will use to access source and target databases.

Agent site configurations

The warehouse agent receives SQL commands from the warehouse server and then passes the commands to the source or target databases.

You can set up the warehouse agent in one of following common configurations:

The warehouse agent, source, and target are located on the same workstation.

The warehouse agent is installed on a system that contains both the warehouse source table and the warehouse target table in the same database, as shown in the following figure.

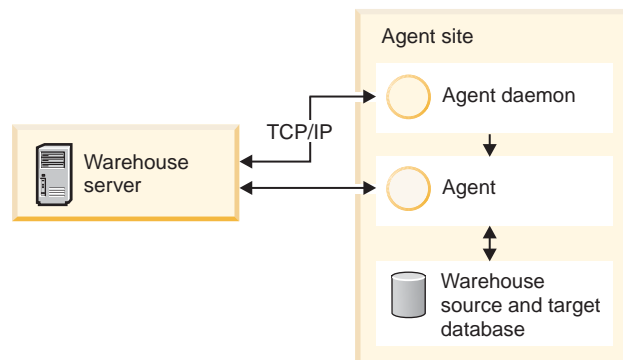


Figure 4. The agent, source, and target are located on the same system.

The warehouse server can also be located on the same system as the warehouse agent, warehouse source, and warehouse target.

The warehouse agent and warehouse target are located on the same system.

The warehouse agent is installed on the system that contains the warehouse target table.

Setting up your warehouse

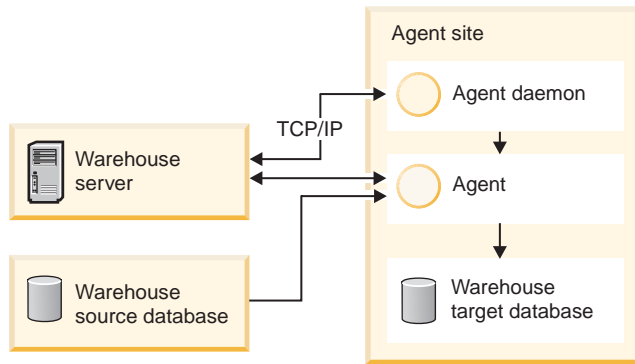


Figure 5. The agent and target are on the same system

This configuration offers the best performance when the source and target reside on different systems.

The warehouse agent and warehouse source are located on the same system.

The warehouse agent is installed on the system that contains the warehouse source, as shown in following figure.

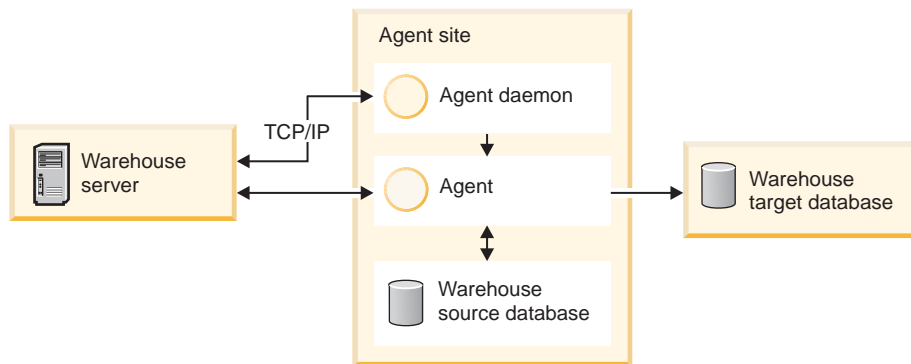


Figure 6. The warehouse agent and source on the same system.

Related concepts:

- “Agent sites for user-defined programs” on page 190

Related tasks:

- “Defining an agent site” on page 18
- “Updating your environment variables on z/OS” on page 19
- “Adding or deleting administrative interfaces and warehouse agents in the Data Warehouse Center” on page 218

Defining an agent site

The Data Warehouse Center uses the local agent as the default agent for all Data Warehouse Center activities. However, you might want to use a warehouse agent on a different site from the workstation that contains the warehouse server. You must define the *agent site* to the Data Warehouse Center. The agent site is the workstation where the agent is installed. The Data Warehouse Center uses this definition to identify the workstation on which to start the agent.

Procedure:

To use a warehouse agent, you must complete the following tasks:

1. Determine where to place the warehouse agent.
2. Establish connectivity to local and remote databases.
3. For a zSeries warehouse agent, configure TCP/IP, set environment variables, and set up authorization for the warehouse agent daemon.
4. Start the agent daemon.
5. Verify communication between the warehouse server and the warehouse agent.
6. Define the agent site to the Data Warehouse Center.

Related concepts:

- “Agent site configurations” on page 17
- “Agent sites for user-defined programs” on page 190

Updating your environment variables on z/OS

You might need to update your environment variables.

Prerequisites:

You must be logged on as the user who starts the agent daemon.

Procedure:

To update your environment variables, add the following variables to the .profile file in your home directory:

Table 2. Variables for the .profile file

Variable

```
export VWS_LOGGING=/usr/lpp/DWC/logs
export VWP_LOG=/usr/lpp/DWC/vwp.log
export VWS_TEMPLATES=/usr/lpp/DWC/templates
export DSNA0INI=/usr/lpp/DWC/inisamp
export LIBPATH=/usr/lpp/DWC/:$LIBPATH
export PATH=/usr/lpp/DWC/:$PATH
export STEPLIB=DSN710.SDSNEXIT:DSN710.SDSNLOAD
```

If you want to receive ODBC (CLI) traces, add the following variables to your ODBC initialization file:

Table 3. Variables for the ODBC initialization file

Variable

```
APPLTRACE
APPLTRACEFILENAME
DIAGTRACE
TRACEFILENAME
```

Data Warehouse Center security

Because the Data Warehouse Center stores user IDs and passwords for various databases and systems, the Data Warehouse Center security structure is separate from the database and operating system security. This structure consists of warehouse groups and warehouse users. Users gain privileges and access to Data Warehouse Center objects by belonging to a warehouse group. A *warehouse group* is a named grouping of warehouse users and *privileges*, which give users the authorization to perform functions. Warehouse users and warehouse groups do not need to match the database users and database groups that are defined for the warehouse control database.

During initialization, you specify the ODBC name of the warehouse control database, a valid DB2[®] user ID, and a password. The Data Warehouse Center authorizes this user ID and password to update the warehouse control database. In the Data Warehouse Center, this user ID is defined as the *default warehouse user*. The default warehouse user requires a different type of database and operating system authorization for each operating system that the warehouse control database supports.

When you log on to the Data Warehouse Center, the Data Warehouse Center verifies that you are authorized to open the Data Warehouse Center administrative interface by comparing your user ID to the defined warehouse users.

If you don't want to define security, you can log on as the default warehouse user and access all Data Warehouse Center objects and perform all Data Warehouse Center functions. The default warehouse user is a part of the default warehouse group. This warehouse group has access to all the objects that are defined in the Data Warehouse Center, unless you remove objects from the warehouse group.

However, you probably want different groups of users to have different access to objects within the Data Warehouse Center. For example, warehouse sources and warehouse targets contain the user IDs and passwords for their corresponding databases. You might want to restrict access to the warehouse sources and warehouse targets that contain sensitive data, such as personnel data.

You restrict the actions that users can perform by assigning privileges to the warehouse group. In the Data Warehouse Center, two privileges can be assigned to groups: administration privilege and operations privilege.

Administration privilege

Users in the warehouse group can define and change warehouse users and warehouse groups, change Data Warehouse Center properties, import metadata, and define which warehouse groups have access to objects when they are created.

Operations privilege

Users in the warehouse group can monitor the status of scheduled processing.

Privileges are assigned (by a user with administration privilege) to groups. In order for warehouse users to have a privilege, they must belong to a warehouse group that has the privilege.

In addition to the privileges, a warehouse group contains lists of objects to which the users in the group have access. You can specify access to sources, targets, and processes.

|
| For example, you might define a warehouse user, such as CHenry, that
| corresponds to someone who uses the Data Warehouse Center. You might then
| define a warehouse group, such as Shoppers, that is authorized to access certain
| warehouse sources, and add CHenry to this new warehouse group. CHenry is
| authorized to access the warehouse sources that are included in the Shoppers
| group.

You can give users various types of authorization. You can include any of the different types of authorization in a warehouse group. You can also include a warehouse user in more than one warehouse group. The combination of the groups to which a user belongs is the user's overall authorization.

When a user defines a new object to the Data Warehouse Center and does not have administration privilege, all of the groups to which the user belongs will have access to the new object by default. The list of groups to which they can assign access is limited to the groups to which they belong. The Security page of the object notebook will not be available to the user.

The list of tables or views that users can access from a source will be limited by their group membership as well, so that they will be able to choose from among the tables and views to which they have access. The set of actions available to the user through the Data Warehouse Center will be limited by the level of security that the user has. For example, a user will not be able to access the properties of an object if the user does not belong to a group that has access to the object.

The Data Warehouse Center works with the security for your database manager by including the user ID and password for the database as part of the warehouse source and warehouse target properties.

The following figure shows the relationship between warehouse users, warehouse groups, and the user IDs and passwords for the warehouse databases:

Setting up your warehouse

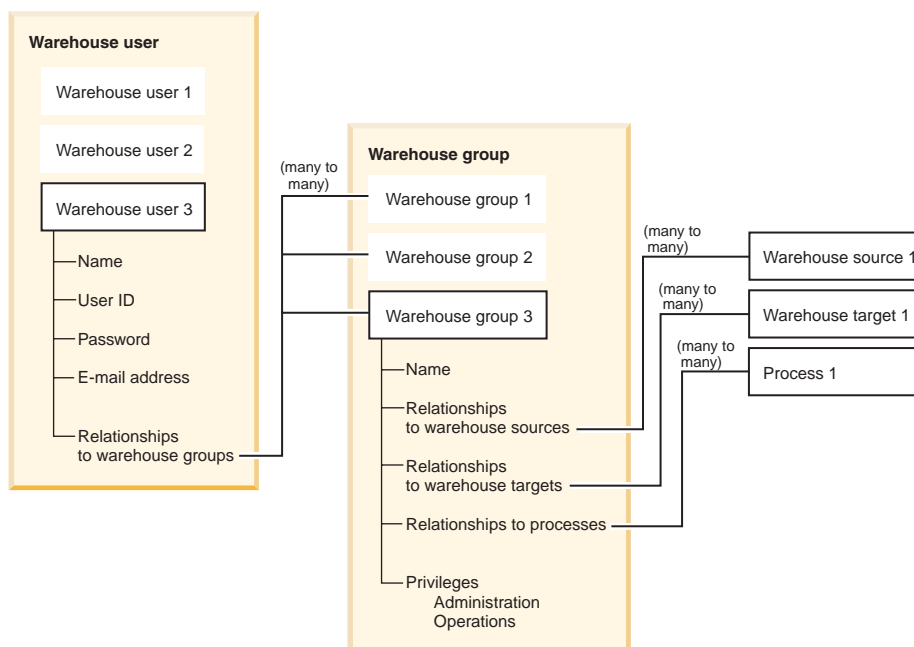


Figure 7. Relationship between warehouse users, warehouse groups, and the user IDs and passwords for the warehouse databases

Related concepts:

- “Data warehouse objects” on page 2

Related tasks:

- “Defining warehouse security” on page 22

Defining warehouse security

The Data Warehouse Center controls access with user IDs. When a user logs on, the user ID is compared to the warehouse users that are defined in the Data Warehouse Center to determine whether the user is authorized to access the Data Warehouse Center. You can authorize additional users to access the Data Warehouse Center by defining new warehouse users.

The user ID for the new user does not require authorization to the operating system or the warehouse control database. The user ID exists only within the Data Warehouse Center.

In the Data Warehouse Center, warehouse groups include the authorization to perform certain tasks and access objects in the Data Warehouse Center. To authorize one or more users to perform tasks, you must define a warehouse group, and then add the users to the group.

Procedure:

To define security for your data warehouse:

1. Define warehouse users.
2. Define warehouse groups.

Related concepts:

- “Data Warehouse Center security” on page 20

Setting up your warehouse

Chapter 3. Setting up DB2 warehouse sources

Before you can create steps that access your data sources, you must perform the following tasks:

- Determine which data sources you will use in your warehouse.
- Set up connectivity between the sources and the warehouse agent that you want to use.
- Define the sources to the Data Warehouse Center as warehouse sources.

This chapter describes the DB2 data sources that the Data Warehouse Center can use and tells you how to set up access to them.

Supported DB2 data sources

The Data Warehouse Center supports a wide variety of relational and nonrelational data sources. The following table shows the version and release levels that are supported for DB2 data sources.

Table 4. Supported DB2 data sources

Source database	Version/release
DB2 Universal Database for the following operating systems: <ul style="list-style-type: none">• Windows NT• AIX• Linux• Solaris Operating Environment	5.2 through 8.1
DB2 Universal Database for OS/2	7.1 and 7.2
DB2 Universal Database for Windows 2000	6.1, 7 and 8.1
DB2 Universal Database for Windows XP	7.2 Fixpak 4 or later
DB2 Universal Database Enterprise Server Edition	8.1
DB2 Universal Database Enterprise-Extended Edition	5.2 through 7.2
DB2 Universal Database for iSeries	5.1 or later
DB2 Universal Database for z/OS	5.1 or later
DB2 for VM	5.3.4 or later
DB2 for VSE	8.1

You can access other databases and files, such as workstation-based spreadsheets and Lotus Notes databases, using the Data Warehouse Center generic ODBC driver.

Schema, table, and column names of remote sources in Japanese cannot contain certain characters. Unicode mapping differences may cause the names to be null. For more information, see <http://www.ingrid.org/java/i18n/encoding/ja-conv.html>.

Related concepts:

- “Supported warehouse targets” on page 84

Setting up DB2 warehouse sources

Related reference:

- “Supported non-DB2 data sources” on page 39
- “Column size truncation for non-DB2 warehouse sources” on page 40

Warehouse agent support for DB2 sources

The following warehouse agents support DB2 sources.

- Windows warehouse agent
- AIX warehouse agent
- Linux warehouse agent
- Solaris Operating Environment warehouse agent
- iSeries warehouse agent
- zSeries warehouse agent

Setting up connectivity for DB2 sources (Windows NT, Windows 2000, Windows XP)

The following sections describe how to set up connectivity for DB2 sources on Windows NT, Windows 2000, or Windows XP.

Setting up connectivity for DB2 Universal Database databases (Windows)

When you set up a data warehouse to use a DB2 Universal Database source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

DB2 Universal Database Version 8 server or DB2 client

Procedure:

To set up connectivity for DB2 Universal Database:

1. Install the DB2 server or a DB2 client on the agent site.
2. Catalog the local or remote database as an ODBC system Data Source Name.
3. Identify the user ID and password that can access the source database.
4. Bind database utilities and ODBC(CLI) to the database.

Setting up connectivity for DB2 DRDA databases (Windows)

When you set up a data warehouse to use a DB2 DRDA database, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Client enabler program DB2 Connect®

Procedure:

To set up connectivity for DB2 DRDA databases:

1. Install DB2 Connect on the gateway site.
2. Catalog the node of the gateway site on the agent site.
3. Catalog the DB2 Connect database on the agent site.
4. Identify the user ID and password that can access to the source database.

Setting up connectivity for DB2 sources (AIX)

The following sections describe how to set up connectivity for DB2 sources on AIX.

Setting up connectivity for a DB2 Universal Database source (AIX)

When you set up a data warehouse to use a DB2 Universal Database source, you must set up connectivity to it.

Prerequisites:

Source/agent connection ODBC

Client enabler program DB2 Universal Database Version 8 server or DB2 client

Restrictions:

If you are using a Data Warehouse Center AIX agent that is linked to access Data Warehouse Center ODBC sources and will access DB2 databases, change the value of the `Driver=` attribute in the DB2 source section of the `.odbc.ini` file as follows:

```
Driver=/usr/opt/db2_08_01/lib/libdb2.so
```

The following example is a sample ODBC source entry for AIX:

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/lib/libdb2.so
Description=DB2 ODBC Database
Database=SAMPLE
```

Procedure:

To set up connectivity for a DB2 Universal Database source:

1. Install the DB2 server or a DB2 client on the agent site.
2. Catalog the remote database.
3. Identify the user ID and password that have access to the source database.
4. Add entries to the `.odbc.ini` file if you are using the ODBC version of the warehouse agent.

Setting up connectivity for a DB2 DRDA database source (AIX)

When you set up a data warehouse to use a DB2 DRDA database source, you must set up connectivity to it.

Prerequisites:

Source/agent connection
ODBC

Client enabler program
DB2 Connect

Restrictions:

If you are using a Data Warehouse Center AIX agent that is linked to access Data Warehouse Center ODBC sources and will access DB2 databases, change the value of the `Driver=` attribute in the DB2 source section of the `.odbc.ini` file as follows:

```
Driver=/usr/opt/db2_08_01/lib/db2_36.o
```

The following example is a sample ODBC source entry for AIX:

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/lib/db2_36.o
Description=DB2 ODBC Database
Database=SAMPLE
```

Procedure:

To set up connectivity for a DB2 DRDA database source:

1. Install DB2 Connect on the gateway site.
2. Catalog the node of the gateway site on the agent site.
3. Catalog the DB2 Connect database on the agent site.
4. Identify the user ID and password that can access to the source database.

Setting up connectivity for DB2 sources (Solaris Operating Environment, Linux)

The following sections describe how to set up connectivity for DB2 sources on the Solaris Operating Environment and Linux.

Setting up connectivity for a DB2 Universal Database source (Solaris Operating Environment, Linux)

When you set up a data warehouse to use a DB2 Universal Database source, you must set up connectivity to it.

Prerequisites:

Source/agent connection
ODBC

Client enabler program
DB2 Universal Database Version 8 server or DB2 client

Restrictions:

If you are using a Data Warehouse Center agent for the Solaris Operating Environment or Linux that is linked to access Data Warehouse Center ODBC sources and will access DB2 databases, change the value of the `Driver=` attribute in the DB2 source section of the `.odbc.ini` file as follows:

```
driver=/opt/IBM/db2/V8.1/lib/libdb2.so
##Driver=/opt/IBM/db2/V8.1/lib/libdb2_36.so.1
```

The following example is a sample ODBC source entry for the Solaris Operating Environment and Linux:

```
[SAMPLE]
Driver=/opt/IBM/db2/V8.1/lib/libdb2_36.so.1
Description=SAMPLE DB2 ODBC Database
Database=SAMPLE
```

Procedure:

To set up connectivity for a DB2 Universal Database source:

1. Install the DB2 server or a DB2 client on the agent site.
2. Catalog the remote database.
3. Specify the user ID and password that have access to the source database.
4. Add entries to the `.odbc.ini` file if you are using the ODBC version of the warehouse agent.

Setting up connectivity for a DRDA data source (Solaris Operating Environment, Linux)

When you set up a data warehouse to use a DB2 DRDA source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

DB2 Connect

If you are using a Data Warehouse Center agent for the Solaris Operating Environment or Linux that is linked to access Data Warehouse Center ODBC sources and will access DB2 databases, change the value of the `Driver=` attribute in the DB2 source section of the `.odbc.ini` file as follows:

```
Driver=/opt/IBMd2/V8.1/lib/libdb2_36.so
##Driver=/opt/IBMd2/V8.1/lib/libdb2_36.so.1
```

The following example is a sample ODBC source entry for the Solaris Operating Environment and Linux:

```
[SAMPLE]
Driver=/opt/IBMd2/V8.1/lib/libdb2_36.so
Description=Text driver
#optional:
#Database=/home/db2inst4/A11Dtype.txt
```

Procedure:

To set up connectivity for a DRDA data source:

1. Install DB2 Connect on the gateway site.

Setting up DB2 warehouse sources

2. Catalog the node of the gateway site on the agent site.
3. Catalog the DB2 Connect database on the agent site.
4. Add entries to the .odbc.ini file if you are using the ODBC version of the warehouse agent.

Setting up connectivity for DB2 sources (OS/2)

The following sections describe how to set up connectivity for DB2 sources on OS/2.

Setting up connectivity for a DB2 Universal Database source (OS/2)

When you set up a data warehouse to use a DB2 Universal Database source, you must set up connectivity to it.

Procedure:

To set up connectivity for a DB2 Universal Database source:

1. Install the DB2 server or a DB2 client on the agent site.
2. Catalog the remote database.
3. Identify the user ID and password with access to the source database.
4. Bind database utilities and ODBC(CLI) to the database.

Setting up connectivity for a DRDA data source (OS/2)

When you set up a data warehouse to use a DB2 DRDA source, you must set up connectivity to it.

Prerequisites:

Source/agent connection
ODBC

Procedure:

To set up connectivity for a DRDA data source:

1. Install DB2 Connect on the gateway site.
2. Catalog the node of the gateway site on the agent site.
3. Catalog the DB2 Connect database on the agent site.
4. Identify the user ID and password with access to the source database.

Accessing source databases

| The following topics describe how to set up access to source databases. To set up
| access to DB2 data sources you need to define privileges and establish connectivity
| to the source databases. You can use any DB2 database as a source database for
| your warehouse.

Defining privileges to DB2 data sources

Defining privileges to a DB2 data source is the first step in setting up access.

Procedure:

To define privileges to DB2 data sources:

1. Set up a user ID with the following privileges on the source system:
 - BINDADD (database-level privilege)
 - CONNECT (database-level privilege)
2. Give an explicit SELECT privilege to the following SYSIBM system tables:
 - SYSIBM.SYSTABLES
 - SYSIBM.SYSCOLUMNS
 - SYSIBM.SYSDBAUTH
 - SYSIBM.SYSTABAUTH
 - SYSIBM.SYSINDEXES
 - SYSIBM.SYSRELS
 - SYSIBM.SYSTABCONST
3. Optional: Give the explicit SELECT privilege to any other tables that you want to access.

After you define privileges to a DB2 data source, you must establish connectivity to it.

Establishing connectivity to DB2 data sources

After the system administrator defines the required privileges, establish connectivity to the source database at the agent site.

Procedure:

To establish connectivity:

1. If the database is remote, set up communications to the database and catalog the node.
2. Catalog the database.
3. If you are using Windows NT, Windows 2000, Windows XP or the version of the AIX, Solaris Operating Environment, or Linux warehouse agent that uses ODBC, register the database as an ODBC system DSN. If you are using the AIX, zSeries, Solaris Operating Environment, or Linux warehouse agent that uses the CLI interface, catalog the database using DB2 catalog utilities.
4. Bind database utilities and ODBC(CLI) to the database. Each type of client requires only one bind.

Setting up access to DB2 DRDA data sources

You can access DB2 DRDA source databases through a gateway or by establishing a direct connection.

Prerequisites:

Setting up DB2 warehouse sources

Use a gateway site to access data from one of the following source databases. Configure the site for DRDA.

- DB2 Universal Database for iSeries®
- DB2 Universal Database for z/OS
- DB2 for VM
- DB2 for VSE

Procedure:

To access a DB2 DRDA source using a gateway site:

1. Verify that the gateway server has the prerequisite products installed.
2. Define privileges to the source databases.
3. Set up the DB2 Connect gateway site.
4. Establish connectivity to the source databases.

Defining privileges to DB2 DRDA data sources

To set up access to a DB2 DRDA source database, you must define privileges to the DB2 DRDA source database.

Procedure:

The system administrator of the source system must set up a user ID with the following privileges on a server that is configured for DRDA:

- For all DRDA servers, the user ID must be authorized to CONNECT to the database.

Additionally, the following system tables, and any tables you want to access, require explicit SELECT privilege:

- SYSIBM.SYSTABLES
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSDBAUTH
- SYSIBM.SYSTABAUTH
- SYSIBM.SYSINDEXES
- SYSIBM.SYSRELS
- SYSIBM.SYSTABCONST
- For DB2 Universal Database for z/OS the user ID must have one of the following authorizations:
 - SYSADM
 - SYSCtrl
 - BINDADD and the CREATE IN COLLECTION NULLID authorization
- For DB2 for VSE or DB2 for VM, the user ID must have DBA authority. To use the GRANT option on the BIND command, the NULLID user ID must have the authority to grant authority to other users on the following tables:
 - SYSTEM.SYSCATALOG
 - SYSTEM.SYSCOLUMNS
 - SYSTEM.SYSINDEXES
 - SYSTEM.SYSTABAUTH
 - SYSTEM.SYSKEYCOLS

- SYSTEM.SYSSYNONYMS
- SYSTEM.SYSKEYS
- SYSTEM.SYSCOLAUTH
- For DB2 Universal Database for iSeries, the user ID must have CHANGE authority or higher on the NULLID collection

After your user ID is set up with the required privileges, you can set up the DB2 Connect gateway site.

Setting up a DB2 Connect gateway site (Windows)

A gateway site is required if you want to connect to a DB2 for iSeries, DB2 for z/OS, DB2 for VM, or DB2 for VSE database source.

Prerequisites:

To set up a DB2 Connect gateway site, you need a user ID on the source system with the privileges that are required for accessing the DB2 DRDA database source that you want to access.

Procedure:

To set up a DB2 Connect gateway site.

1. Install Microsoft SNA Server.
2. Install DB2 Connect.
3. Configure the DB2 Connect system to communicate with the source databases.
4. Update the DB2 node directory, system database directory, and DCS directory.

After you set up the gateway site, you can establish connectivity to a DB2 DRDA source database.

Connecting to DB2 DRDA data sources

After you set up the DB2 Connect gateway site, you can establish connectivity to the following DB2 DRDA source databases:

- DB2 for iSeries
- DB2 for z/OS
- DB2 for VM
- DB2 for VSE

For other DB2 DRDA source databases, you can establish a direct connection without going through a gateway site. The following procedure explains how to establish connectivity in either case.

Prerequisites:

If you are connecting to a DB2 DRDA source database through a gateway site, you must have a user ID on the source system with the required privileges, and you must set up a gateway site.

Procedure:

Setting up DB2 warehouse sources

To establish connectivity to a DB2 DRDA source database at the warehouse agent site:

1. Set up communications to the DB2 Connect site.
2. Catalog the node of the DB2 Connect site.
3. Catalog the database.
4. Register the database as a system DSN by using the ODBC Administrator.
5. Bind the DB2 Connect utilities to the DRDA server. Each type of client requires only one bind.

On Windows you can use the DB2 UDB Configuration Assistant to complete this task.

Related tasks:

- “Setting up access to DB2 DRDA data sources” on page 31
- “Setting up a DB2 Connect gateway site (Windows)” on page 33

Connectivity requirements for remote databases

The following sections describe connectivity requirements for databases that are remote from the iSeries and zSeries warehouse agents.

Connectivity requirements for remote databases from the iSeries warehouse agent

You can access remote databases with the iSeries warehouse agent through Systems Network Architecture (SNA) or IP connectivity that uses IBM Distributed Relational Database Architecture (DRDA), or DRDA over TCP/IP.

You must have DRDA connectivity to access the following remote databases:

- DB2 Universal Database for iSeries
- DB2 Universal Database for z/OS

You can connect from the iSeries warehouse agent to a remote database when the following conditions are met:

- The SNA or IP connection to the remote database is correct.
- The remote database is cataloged in the iSeries Relational Database directory.

You can connect to a remote database from the Data Warehouse Center and query it if the following conditions are met:

- You can connect to the remote database from the iSeries warehouse agent.
- You can query the remote database from the iSeries interactive SQL facility (STRSQL).

Related concepts:

- “Example of how to catalog local and remote database names for the iSeries warehouse agent” on page 36

Related tasks:

- “Setting up access to local and remote databases from the iSeries warehouse agent” on page 35
- “Viewing and editing remote relational database directory entries” on page 36

Setting up access to local and remote databases from the iSeries warehouse agent

You must catalog the names of local and remote databases that you plan to use as warehouse sources or targets in the iSeries Relational Database directory on your agent site. You must also catalog these database names on the remote workstation that your agent accesses.

Prerequisites:

The local database name that you catalog on the agent site must be cataloged as the remote database name on the remote workstation that your agent will access. Also, the remote database name that you catalog on your agent site must be cataloged as the local database name on the remote workstation that the agent will access.

If your source database and target database are located on the same workstation, you must catalog one as local and the other as remote.

Procedure:

To set up access to local or remote databases, add a database name entry to the iSeries Relational Database directory by entering the following command at an iSeries command prompt:

```
ADDRDBDIRE databasename locationname
```

databasename

The name of your iSeries database.

locationname

The location name of your iSeries workstation. You must specify whether the database is local or remote.

You must supply both the name of the database and the name of the location, even if they are the same name.

For the local database, the location name is the *LOCAL keyword. For each remote database, the location field must contain the SNA LU name.

If you change the name of a database in the Remote Database directory, you must update each warehouse source that refers to it. Failure to do so will result in warehouse source database connection errors.

Related concepts:

- “Connectivity requirements for the warehouse server and the warehouse agent” on page 11
- “Example of how to catalog local and remote database names for the iSeries warehouse agent” on page 36

Related tasks:

- “Viewing and editing remote relational database directory entries” on page 36

Related reference:

- “Connectivity requirements for remote databases from the iSeries warehouse agent” on page 34

Example of how to catalog local and remote database names for the iSeries warehouse agent

Sabir is creating a data warehouse. He wants to catalog the database names for a database that is named Sales and a database that is named Expenses. The database named Sales is located on the same workstation as the iSeries™ agent. The database named Expenses is located on the remote workstation that the agent will access. The following table shows how Sabir should catalog each database on each workstation.

Table 5. Example of how Sabir catalogs each database

Database name	Location	Cataloged as local or remote on agent site	Cataloged as local or remote on remote machine
Sales	Agent site	Local	Remote
Expenses	Machine agent accesses	Remote	Local

Viewing and editing remote relational database directory entries

You can use the WRKRDBDIRE command to view, add, change, and remove remote relational database directory entries.

Prerequisites:

The local database name that you catalog on your agent site must be cataloged as the remote database name on the remote workstation that your agent will access. Also, the remote database name that you catalog on your agent site must be cataloged as the local database name on the remote workstation your agent will access.

If your source database and target database are located on the same workstation, you must catalog one as local and the other as remote.

Procedure:

To view, add, change, and remove remote relational database directory entries, enter the following command at an iSeries command prompt:

```
WRKRDBDIRE
```

A list of currently defined remote database names is displayed. A set of options is displayed at the top of the window.

Requirements for accessing remote data sources with the zSeries warehouse agent

You can access remote databases from the zSeries® agent using IBM® Distributed Relational Database Architecture™ (DRDA) over TCP/IP.

DRDA® connectivity is required to access the following remote databases:

- DB2® Universal Database for z/OS™

- DB2 Universal Database™ for iSeries™
- Oracle®, Sybase, Informix®, and other similar database products, when DB2 Information Integrator is used

Requirements for accessing relational data sources with the zSeries warehouse agent

When you use the zSeries® agent to access relational data sources, Systems Network Architecture (SNA) LU 6.2 is required as the communications protocol. Although the DB2® federated server supports TCP/IP as an application requester, it does not have an application server. Also, the DB2 federated server can not be used as a target for z/OS™, because federated systems do not support two-phase commit from DRDA®, a requirement of DB2 Universal Database for z/OS.

DB2 warehouse sources

To determine which tables in the data source that you want to use, you can use the Sample Contents function to view the data in the source tables. You view the data from one table at a time. The Data Warehouse Center displays all the column names of the table, regardless of whether data is present in the column. A maximum of 200 rows of data can be displayed.

You can view the data before or after you import the definition of the table.

Any warehouse user can define a warehouse source, but only warehouse users who belong to a warehouse group with access to the warehouse source can change it.

Ordinary identifiers

The Data Warehouse Center supports source tables that use ordinary SQL identifiers. An ordinary identifier:

- Must start with a letter
- Can include uppercase letters, numbers, and underscore characters
- Cannot be a reserved word

If a table has a lowercase letter as part of its ordinary identifier, the Data Warehouse Center stores the lowercase letter as an uppercase letter.

Delimited identifiers

The Data Warehouse Center supports source tables that use delimited identifiers. A delimited identifier:

- Is enclosed within quotation marks
- Can include uppercase and lowercase letters, numbers, underscore characters, and spaces
- Can contain a double quotation mark, represented by two consecutive quotation marks: ""

Metadata from tables

To save time, you can import metadata from certain types of tables, files, and views into the Data Warehouse Center. Importing the metadata saves you the time it would take to define the sources manually.

Agent sites

If more than one agent site is specified in the warehouse source, the warehouse server uses the agent site with the name that appears first according to the user's locale for the import process.

For example, your warehouse source has three agent sites selected: Default agent, AIX[®] agent, and zSeries[®] agent. The warehouse server uses the AIX agent site for the import process.

Related tasks:

- "Defining warehouse sources for use with a DB2 federated system" on page 60
- "Defining DB2 warehouse sources" on page 38

Related reference:

- "Column size truncation for non-DB2 warehouse sources" on page 40

Defining DB2 warehouse sources

After you establish connectivity to your source and determine which source tables you want to use, you can define a DB2 source database in the Data Warehouse Center.

Prerequisites:

If you are using source databases that are remote to the warehouse agent, you must register the databases on the workstation that contains the warehouse agent.

When you define a warehouse source for a **DB2 for VM** database, which is accessed through a DRDA gateway, the following restrictions apply to the use of CLOB and BLOB data types:

- You cannot use the Sample Contents function to view data of CLOB and BLOB data types.
- You cannot use columns of CLOB and BLOB data types with an SQL step. This restriction applies to the DB2 for VM Version 5.2 server in which LOB objects cannot be transmitted using DRDA to a DB2 Version 8 client.

Procedure:

To define a DB2 source in the Data Warehouse Center:

1. In the Data Warehouse Center, open the Define Warehouse Source notebook.
2. Add information about the warehouse source.
3. Specify information about the source database.
4. Import source tables and views into the warehouse source.
5. Authorize warehouse groups to access a warehouse source.
6. Specify an agent site to access a warehouse source.

Related tasks:

- "Defining a warehouse source based on relational databases: Data Warehouse Center help"

Chapter 4. Setting up non-DB2 warehouse sources

Before you create steps that access a non-DB2 data source, you must perform the following tasks:

- Determine which non-DB2 data sources you will use in your warehouse.
- Set up connectivity between the non-DB2 data sources and the warehouse agent that you plan to use.
- Define the non-DB2 data sources to the Data Warehouse Center as warehouse sources.

This chapter describes the non-DB2 data sources with which the Data Warehouse Center works and tells you how to set up access to them.

Supported non-DB2 data sources

The following table lists the non-DB2 data sources that are supported by the Data Warehouse Center.

Table 6. Supported non-DB2 data sources

Source database	AIX, Linux, Solaris Operating Environment, Windows agents
Informix Dynamic Server with Universal Data Option (without client)	9.1.4 and later
Informix Dynamic Server (without client)	9.2x and 9.3x
Oracle (without client)	8.1.6, 8.1.7, 9.0.1, and 9.2
Microsoft® SQL Server (without client)	7.0 and 2000
Microsoft Excel	5.1 and 7.0; supported only on Windows environments
Microsoft Access	2000 and 2002; supported only on Windows environments
Red Brick Warehouse	6.20.xC4
Sybase Adaptive Server (without client)	11.5 and later
Sybase Adaptive Server Enterprise (without client)	12.0 and 12.5

Red Brick Warehouse support is provided through warehouse-supplied programs in the DB2 Warehouse Manager Standard Edition. ODBC and Red Brick Client setup are required to use some of the functions provided by these programs. You do not define a warehouse source or a warehouse target for the Red Brick database.

The Data Warehouse Center includes several DataDirect Technologies, Inc., ODBC drivers for accessing non-DB2 data. The Data Warehouse Center Version 8.2 includes the DataDirect Version 4.2 Driver Manager.

Setting up non-DB2 warehouse sources

Schema, table, and column names of remote sources in Japanese cannot contain certain characters. Unicode mapping differences may cause the names to be null. For more information, see <http://www.ingrid.org/java/i18n/encoding/ja-conv.html>.

Warehouse agent support for non-DB2 sources

The following table shows which warehouse agents support which sources.

Table 7. Sources that are supported by warehouse agents

Data Source	Windows warehouse agent	AIX or Solaris Operating Environment warehouse agent	Linux warehouse agent	zSeries warehouse agent
DB2 Information Integrator	✓	✓		✓
Oracle	✓	✓	✓	
Sybase Adaptive Server	✓	✓	✓	
Microsoft SQL Server	✓	✓	✓	
Informix	✓	✓	✓	
Generic ODBC driver	✓	✓	✓	

Related tasks:

- “Defining DB2 warehouse sources” on page 38

Related reference:

- “Warehouse agent support for DB2 sources” on page 26

Column size truncation for non-DB2 warehouse sources

The Data Warehouse Center will truncate the column size of the following data types to 128 KB if the column size is 128 KB or greater.

- LONG VAR CHAR
- LONG VAR BINARY
- MEMO FIELD
- NTEXT

To avoid having the column size truncated, restrict the size of the data in these column types to 128 KB or less.

Related tasks:

- “Defining a non-DB2 warehouse source in the Data Warehouse Center” on page 58
- “Specifying database information for a non-DB2 warehouse source in the Data Warehouse Center” on page 59

Setting up connectivity for non-DB2 sources (Windows)

The following sections describe how to set up connectivity for non-DB2 sources on Windows NT, Windows 2000 and Windows XP.

Setting up connectivity for a Sybase Adaptive Server source – without client (Windows)

When you set up a data warehouse to use a Sybase Adaptive Server source, you must set up connectivity to it.

Prerequisites:**Source/agent connection**

ODBC

Client enabler program

DataDirect Version 4.2 Driver Manager and Sybase driver.

Procedure:

To set up connectivity for a Sybase Adaptive Server data source, catalog the remote database, open the ODBC Data Source Administrator and define the properties of the source.

Setting up connectivity for an Oracle source (Windows)

When you set up a data warehouse to use an Oracle source, you must set up connectivity to it.

Prerequisites:**Source/agent connection**

ODBC

Client enabler program

- Oracle SQL*Net V2
- DataDirect Version 4.2 Driver Manager and Oracle driver

Database client requirements

- For Oracle Version 8: To access remote Oracle8 database servers at a level of version 8.0.3 or later, install Oracle Net8 Client version 7.3.4.x, 8.0.4, or later.
- On Intel systems, install the appropriate DLLs for the Oracle Net8 Client (such as Ora804.DLL, PLS804.DLL and OCI.DLL) on your path.

Procedure:

To set up connectivity for an Oracle source:

1. Catalog the remote database according to client enabler instructions.
2. Identify the user ID and password with access to the source database.

Setting up connectivity for an Informix 9.2 source – with client (Windows)

When you set up a data warehouse to use an Informix 9.2 source, you must set up connectivity to it. You are not required to use a client to access an Informix 9.2 source. If you use an Informix client, follow the instructions in this topic for setting up connectivity. If you do not use a client, follow the instructions for setting up connectivity without an Informix client.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

- For Informix 9.x, i-connect 9.x
- DataDirect Driver Manager and Informix driver

Procedure:

To set up connectivity for an Informix source, register the system DSN for the ODBC driver.

Related tasks:

- “Setting up connectivity for an Informix 9.2 source – without client (Windows)” on page 42

Setting up connectivity for an Informix 9.2 source – without client (Windows)

You can connect directly to an Informix 9.2 source without using a client. This topic describes how to set up connectivity for an Informix 9.2 source with no client.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

DataDirect Driver Manager and Informix driver

Procedure:

To set up connectivity for an Informix 9.2 source with no client required, register the system DSN for the ODBC driver.

Setting up connectivity for a Microsoft SQL Server data source (Windows)

When you set up a data warehouse to use a Microsoft SQL Server source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

- For access to a Version 7.0 DBMS, Microsoft SQL Server DB-Library and Net-Library Version 7.0
- DataDirect Version 4.2 Driver Manager and Microsoft SQL Server driver

Procedure:

To set up connectivity for a Microsoft SQL Server data source:

1. Catalog the remote database according to client enabler instructions.
2. Identify the user ID and password that has access to the source database.

Setting up connectivity for a Microsoft Access source or a Microsoft Excel data source (Windows)

When you set up a data warehouse to use a Microsoft Access source or a Microsoft Excel data source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Procedure:

To set up connectivity for a Microsoft Access source or a Microsoft Excel data source, use the generic ODBC connect string. See each product's help topics for a mapping of the ANSI SQL data types that it supports.

Setting up connectivity for an IMS, or a VSAM data source (Windows)

Index entries

When you set up a data warehouse on z/OS to use an IMS or a VSAM source, you must set up connectivity to the source. Depending on whether you are using the zSeries warehouse agent, you can use the CROSS ACCESS ODBC driver or DB2 Information Integrator to establish connectivity.

Prerequisites:

Database access programs

If you are not using the zSeries warehouse agent, use one of the following programs:

- The CROSS ACCESS ODBC driver
- DB2 Information Integrator

Source/agent connection

- If you are using the CROSS ACCESS ODBC driver, use ODBC as your database access program.
- If you are using DB2 Information Integrator, use TCP/IP.

Procedure:

Setting up non-DB2 warehouse sources

To set up connectivity for an IMS or VSAM data source using the ACCESS ODBC driver:

1. Establish a link from the agent site to the host.
2. Install and configure the data server on the host.
3. Identify the user ID and password with access to the source database.

To set up connectivity for an IMS or VSAM data source from a federated database using DB2 Information Integrator:

1. Establish a link from the workstation to the host.
2. Install and configure the adapter on the host.
3. Identify the user ID and password with access to the source database.

To set up connectivity for an IMS or VSAM data source from the agent site:

1. Catalog the node on which the federated server resides.
2. Catalog the federated database.

related links

Managing the warehouse agent for non-DB2 sources

This section describes how to verify which version of the warehouse agent you are using, and how to switch to the ODBC warehouse agent.

Verifying the warehouse agent (AIX, Solaris Operating Environment, Linux)

There are two versions of the warehouse agent for AIX, Solaris Operating Environment, and Linux: one for ODBC access and one for CLI access. The non-DB2 databases supported for AIX require that you install the ODBC version of the warehouse agent. You can verify which version of the warehouse agent is installed.

Procedure:

To verify which version of the warehouse agent is installed, enter the following command at the command prompt:

```
ls -l sqllib/bin/IWH2AGNT
```

- If the command returns IWH2AGNT.db2cli, you are using the DB2 CLI version.
- If the command returns IWH2AGNT.ivodbc, you are using the ODBC version.

Switching between versions of the warehouse agent

There are two versions of the warehouse agent, the DB2 CLI version and the ODBC version. The ODBC version of the warehouse agent is required for connecting to non-DB2 warehouse sources.

Procedure:

To change from the DB2 CLI warehouse agent to the ODBC warehouse agent, enter the following command at a command prompt:

```
IWH.agent.db.interface intersolv
```


To change from the ODBC warehouse to the DB2 CLI warehouse agent, enter the following command at a command prompt:

```
IWH.agent.db.interface db2cli
```

You must restart the warehouse agent daemon only if the IWH.environment variable has been modified to run either the DB2 CLI warehouse agent or the ODBC warehouse agent.

Setting up connectivity for non-DB2 sources (AIX)

The following sections describe how to set up connectivity for non-DB2 sources on AIX.

Setting up connectivity for a Sybase Adaptive Server source – without client (AIX)

When you set up a data warehouse to use a Sybase Adaptive Server (without client) source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

DataDirect Version 4.2 Driver Manager and Sybase driver

The following example is a sample odbc.ini file entry for AIX:

```
[SYBASE_125]
Driver=/usr/opt/db2_08_01/odbc/lib/ibase19.so
Description=Sybase 125 ODBC Database
Database=sample
ServerName=sybase1125
NetworkAddress=yourServer,4100
#Where yourServer is the name of your server and 4100 is the
#port number. This information can be found in the Sybase Interfaces file.
LogonID=xxxxx
Password=xxxxx
InterfacesFile=/public/sdt_lab/sybase/AIX/System11/interfaces
#The InterfacesFile field must point to the installation
#directory for the Sybase Interfaces file.
```

Procedure:

To set up connectivity for a Sybase Adaptive Server source, configure the ODBC driver.

Setting up connectivity for an Oracle source – without client (AIX)

When you set up a data warehouse to use an Oracle source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Setting up non-DB2 warehouse sources

Client enabler program

DataDirect Driver Manager and Oracle driver

Database client requirements

For Oracle Version 8: Oracle8 Net8 and the Oracle8 SQL*Net shared library (built by the genclntsh8 script)

The following example is a sample odbc.ini file entry for AIX:

```
[ORACLE 8]
Driver=/usr/opt/db2_08_01/odbc/lib/ibor819.so
Description=Oracle 8 Database
Database=ORACLE 8
UID=xxxx
PWD=xxxx
SID=orc1
```

Where ora816 is specified in the SID field of Oracle tnsnames.ora file

Procedure:

To set up connectivity for an Oracle source:

1. Install the version of the AIX warehouse agent with ODBC access.
2. Identify the user ID and password with access to the source database.

Setting up connectivity for an Informix 9.2 source – without client (AIX)

You can connect directly to an Informix 9.2 source without using a client. This topic describes how to set up connectivity for an Informix 9.2 source with no client.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

DataDirect Driver Manager and Informix driver. No client required.

Procedure:

To set up connectivity for an Informix 9.2 source with no client required, configure the ODBC driver.

The following example is a sample odbc.ini file entry for AIX:

```
[INFORMIX 92]
Driver=/usr/opt/db2_08_01/odbc/lib/ibinf19.so
Description=Informix Database
Database=infdb1
HostName=yourHostName.stl.ibm.com
LoginID=xxxx
Password=xxxx
ServerName=inf92
PortNumber=1234
```

Setting up connectivity for a Microsoft SQL Server source – without client (AIX)

When you set up a data warehouse to use a Microsoft SQL Server source, you must set up connectivity to it.

Prerequisites:

Source/agent connection
ODBC

Client enabler program
DataDirect Driver Manager and Informix driver

The following example is a sample odbc.ini file entry for AIX:

```
[MSSQL7]
Driver=/usr/opt/db2_08_01/odbc/lib/ibmsss16.so
Description=Microsoft SQL Server Database
Address=xyyy.zzz.ibm.com
AnsiNPW=yes
Database=test7
UID=xxxx
DESCRIPTION=MicrosoftSQLSenderDatabase
PWD=xxxx
QuotedID=no
```

Procedure:

To set up connectivity for a Microsoft SQL Server source:

1. Install the version of the AIX warehouse agent with ODBC access.
2. Identify the user ID and password that have access to the source database.

Setting up connectivity for non-DB2 sources (Solaris Operating Environment, Linux)

This section describes how to set up connectivity for non-DB2 sources on the Solaris Operating Environment and Linux.

Setting up connectivity for a Microsoft SQL Server source (Solaris Operating Environment, Linux)

When you set up a data warehouse to use a Microsoft SQL Server source, you must set up connectivity to it.

Prerequisites:

Source/agent connection
ODBC

Client enabler program
DataDirect Driver Manager and the Microsoft SQL Server driver.

Verify the version of the warehouse agent that you have installed before you begin this procedure.

Procedure:

Setting up non-DB2 warehouse sources

To set up connectivity for a Microsoft SQL Server source:

1. Install the version of the Solaris Operating Environment warehouse agent with ODBC access.
2. Identify the user ID and password that have access to the source database.

The following example is a sample `odbc.ini` file entry for the Solaris Operating Environment or Linux:

```
[MSSQL7]
#Driver=/opt/IBM/db2/V8.1/odbc/lib/ibmsss16.so
Address=xyyy.zzz.ibm.com
AnsiNPW=yes
Database=test7
UID=xxxxL
DESCRIPTION=MicrosoftSQLSenderDatabase
PWD=xxxx
QuotedID=no
```

Configuring non-DB2 warehouse sources

This section describes how to configure non-DB2 warehouse sources.

Configuring the ODBC driver for non-DB2 warehouse sources— without client (Windows)

ODBC drivers are used to register the source, target, and warehouse control databases that the Data Warehouse Center accesses.

Prerequisites:

Install the ODBC driver that is required to access the database you plan to use. If you do not have the ODBC driver that is required to access the database, you can get the driver from the DB2 Universal Database CD using the Custom installation choice.

Procedure:

To register a DSN for an ODBC driver:

1. Click **Start** —> **Settings** —> **Control Panel**.
2. Double-click **ODBC Data Sources**.
3. Click the **System DSN** tab.
4. Click **Add**.
5. Select the ODBC driver that you want to register.
6. Click **Finish**.

A driver setup window opens. See your database's help topics for further instructions.

Related concepts:

- "Handling errors for Informix 9.2 – without client" on page 49

Related tasks:

- "Defining a non-DB2 warehouse source in the Data Warehouse Center" on page 58

- “Setting up connectivity for an Informix 9.2 source – without client (Windows)” on page 42

Handling errors for Informix 9.2 – without client

You can turn on logging if you encounter the error, “No transaction available,” for Informix® 9.2 without client.

Procedure:

If you encounter the error, “No transaction available,” and you use the **onbar** command for archiving, enter the following command on the command line to turn on logging:

```
ondblog
```

When you enter this command, you receive information that tells you how to switch your database to logging mode.

If you do not use the **onbar** command for archiving, enter the following command on the command line to turn on logging:

```
ontape
```

If you do not archive to tape, you must verify that the archive tape device is pointing to /dev/null. This location might not be valid for everyone. In the file \$INFORMIXDIR/etc/\$ONCONFIG, check that the device statement looks like: LTAPEDEV /dev/null # Log tape device path. With the database server running, log on to Informix and enter the command **ontape -s -B database_name**, where **database_name** is the name of the database for which you want logging turned on.

Related tasks:

- “Configuring the ODBC driver for non-DB2 warehouse sources– without client (Windows)” on page 48
- “Setting up connectivity for an Informix 9.2 source – without client (Windows)” on page 42
- “Setting up connectivity for an Informix 9.2 source – without client (AIX)” on page 46

Creating a target warehouse database (Windows)

Before you catalog a database in ODBC, you must create a target warehouse database.

Procedure:

To create a target warehouse database:

1. Start the DB2 Control Center by clicking **Start** → **Programs** → **IBM DB2** → **General Administration Tools** → **Control Center**.
2. Right-click the **Databases** folder, and click **Create** → **Database Using Wizard**. The Create Database wizard opens.
3. In the **Database name** field, type the name of the database.
4. From the **Default drive** list, select a drive for the database.
5. Optional: In the **Comments** field, type a description of the database.

Setting up non-DB2 warehouse sources

6. Click **Finish**. All other fields and pages in this wizard are optional. The database is created and is listed in the DB2 Control Center.

After you create the target database, catalog the database in ODBC.

Related tasks:

- “Cataloging a target warehouse database for use with a Microsoft Access source database (Windows)” on page 52

Defining a warehouse that uses the Microsoft Access and warehouse target databases (Windows)

After creating and cataloging the Microsoft Access and warehouse target databases, you can define a warehouse that uses the databases.

Procedure:

To create the Data Warehouse Center definitions for the database that you created:

1. In the Data Warehouse Center, open the Define Warehouse Source notebook.
2. On the Database page:
 - a. In the **Data source name** field, specify the name of your data source.
 - b. In the **User ID** field, type the user ID that will access the data source.
 - c. In the **Password** field, type the password for the user ID that you specified.
 - d. In the **Verify password** field, type the password again.
 - e. Select the **Customize ODBC Connect String** check box.
 - f. In the **ODBC Connect String** field, type any additional parameters.
 - g. On the Agent Sites page, specify the agent site on which you registered the Microsoft Access source database and the DB2 warehouse database.
3. Create a warehouse target for the DB2 database.
4. Import tables from the Microsoft Access source.
5. Create a step that uses one or more source tables from the warehouse source for the Microsoft Access database, and that creates a target table in the DB2 warehouse database.
6. Promote the step to test mode.
7. Run the step by right-clicking the step and clicking **Test**.
8. Verify that the data that you created in the Microsoft Access database is in the warehouse database. Enter the following command in the DB2 Command Line Processor window:

```
select * from prefix.table-name
```

prefix The prefix of the warehouse database (such as IWH).

table-name

The name of the warehouse target table.

You will see the data that you entered in the Microsoft Access database.

Related tasks:

- “Setting up connectivity for a Microsoft Access source or a Microsoft Excel data source (Windows)” on page 43
- “Importing table definitions from a Microsoft Access database (Windows)” on page 51

- “Cataloging a Microsoft Access source database in ODBC (Windows)” on page 51
- “Cataloging a target warehouse database for use with a Microsoft Access source database (Windows)” on page 52

Importing table definitions from a Microsoft Access database (Windows)

You can import table definitions from a Microsoft Access database on Windows operating systems.

Prerequisites:

You must create and catalog a Microsoft Access source and a warehouse target database.

DRDA support for the CLOB data type is required for z/OS and iSeries. The CLOB data type is supported for z/OS beginning with DB2 Universal Database, Version 6.

Restrictions:

The Data Warehouse Center truncates the Memo field of a Microsoft Access database to 128 KB. To support practical system configurations, the Data Warehouse Center truncates the values exceeding 128 KB. If you do not want this data truncated, change the data type of the column that is receiving the Memo field data from LONG VARCHAR to CLOB before you use the table in a step.

Procedure:

To import tables from a Microsoft Access database:

1. In the Define Warehouse Sources notebook, click the **Tables and Views** page.
2. Expand the **Tables** folder to import table definitions from your Microsoft Access database.
3. Click **OK** on the Filter window to list all the tables in your database.
4. Verify that the tables that you created in the database are in the **Available Tables** list.
5. Select the table that you want to use from the **Available Tables** list.
6. Click **>**.

The table moves to the **Selected Tables** list.

Cataloging a Microsoft Access source database in ODBC (Windows)

After you create a Microsoft Access database, the next step in setting up access is to catalog the database in ODBC.

Procedure:

To catalog a Microsoft Access database in ODBC:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click **ODBC**.

Setting up non-DB2 warehouse sources

3. Click **System DSN**.
4. Click **Add**.
5. Select **Microsoft Access Driver** from the **Installed ODBC Drivers** list.
6. Click **OK**.
7. Type the database alias in the **Data Source Name** field.
8. Optional: Type a description of the database in the **Description** field.
9. Click **Select**.
10. Select the path and file name of the database from the list boxes.
11. Click **OK**.
12. Click **Advanced**.
13. Type a user ID in the **Login Name** field.
14. Type a password in the **Password** field.
15. In the **Options** list, verify that the value of **FIL** is **MS Access**.
16. Click **OK** in the Set Advanced Options window.
17. Click **OK** in the ODBC Microsoft Access Setup window.
18. Click **Close**.

After you catalog the database in ODBC, create a target warehouse database.

Related tasks:

- “Setting up connectivity for a Microsoft Access source or a Microsoft Excel data source (Windows)” on page 43
- “Cataloging a target warehouse database for use with a Microsoft Access source database (Windows)” on page 52

Cataloging a target warehouse database for use with a Microsoft Access source database (Windows)

After you create a target database, the next step in setting up access to a Microsoft Access source database is to catalog the target database in ODBC.

Procedure:

To catalog a target database in ODBC:

1. Click **Start** —> **Settings** —> **Control Panel**.
2. Double-click **ODBC**.
3. Click **System DSN**.
4. Click **Add**.
5. In the **Installed ODBC Drivers** list, click **IBM DB2 ODBC Driver**.
6. Click **OK**.
7. Type the database alias in the **Data Source Name** field.
8. Optional: Type a description of the database in the **Description** field.
9. Click **Select**.
10. Select the path and file name of the database from the list boxes.
11. Click **OK**.
12. Click **Close**.

After your catalog the target database, define a warehouse that uses the database.

Related tasks:

- “Defining a warehouse target” on page 84

Using a Microsoft Excel spreadsheet as a warehouse source (Windows)

Before you can access a Microsoft Excel source, you must define a warehouse that uses the spreadsheet.

Procedure:

To use a Microsoft Excel spreadsheet as a warehouse source:

1. Create a Microsoft Excel spreadsheet. Note the path and file name of the spreadsheet, because you will use it later.
2. Catalog the spreadsheet in ODBC.
3. Create a warehouse target database.
4. Catalog the warehouse target database.
5. Define the source to the Data Warehouse Center.

Cataloging a Microsoft Excel data source in ODBC (Windows)

After you create a Microsoft Excel spreadsheet to use as a warehouse source, you must catalog the source in ODBC before you can access it from the Data Warehouse Center.

Prerequisites:

If you are using the Microsoft Excel 95/97 ODBC driver to access the Excel spreadsheets, create a named table for each of the worksheets within the spreadsheet.

Procedure:

To catalog a Microsoft Excel spreadsheet in ODBC:

1. Click **Start** —> **Settings** —> **Control Panel**.
2. Double-click **ODBC**.
3. Click **System DSN**.
4. Click **Add**.
5. In the **Installed ODBC Drivers** list, select **Microsoft Excel Driver**.
6. Click **OK**.
7. In the **Data Source Name** field, type the spreadsheet alias.
8. Optional: In the **Description** field, type a description of the spreadsheet.
9. In the **Version** list, select **Excel 97**.
10. Click **Select Workbook**.
11. Select the path and file name of the database from the list boxes.
12. Click **OK**.
13. Click **OK** in the ODBC Microsoft Excel Setup window.
14. Click **Close**.

Related tasks:

Setting up non-DB2 warehouse sources

- “Setting up connectivity for a Microsoft Access source or a Microsoft Excel data source (Windows)” on page 43
- “Using a Microsoft Excel spreadsheet as a warehouse source (Windows)” on page 53

Creating named tables for Microsoft Excel data sources (Windows)

If you are using the Microsoft Excel 95/97 ODBC driver to access the Excel spreadsheets, you need to create a named table for each of the worksheets within the spreadsheet.

Procedure:

To create named tables:

1. Select the columns and rows that you want to include in the table.
2. Click **Insert** → **Name** → **Define**.
3. Verify that the **Refers to** field in the Define Name window contains the cells that you selected in . To include all of the cells that you selected, click the icon on the far right of the **Refers to** field.
4. Type a name (or use the default name) for the marked data.
5. Click **OK**.

You can now import tables when you define your warehouse source without selecting the **Include system tables** check box.

Related tasks:

- “Setting up connectivity for a Microsoft Access source or a Microsoft Excel data source (Windows)” on page 43
- “Using a Microsoft Excel spreadsheet as a warehouse source (Windows)” on page 53
- “Cataloging a Microsoft Excel data source in ODBC (Windows)” on page 53
- “Creating a target warehouse database for use with a Microsoft Excel data source (Windows)” on page 54
- “Cataloging a target warehouse database for use with a Microsoft Excel data source (Windows)” on page 55
- “Defining sources and targets to the Data Warehouse Center that use a Microsoft Excel data source (Windows)” on page 55

Creating a target warehouse database for use with a Microsoft Excel data source (Windows)

After you create and catalog a Microsoft Excel source, create a warehouse target database.

Procedure:

To create a warehouse target database for use with a Microsoft Excel source:

1. Start the DB2 Control Center by clicking **Start** → **Programs** → **IBM DB2** → **General Administration Tools** → **Control Center**.

2. Right-click the **Databases** folder, and click **Create** —> **Database Using Wizard**. The Create Database wizard opens.
3. In the **Database name** field, type the name of the database.
4. In the **Default drive** list, select a drive for the database.
5. In the **Comments** field, type a description of the database.
6. Click **Finish**. All other fields and pages in this wizard are optional. The database is created and is listed in the DB2 Control Center.

After you create the target database, catalog the database in ODBC.

Related tasks:

- “Cataloging a target warehouse database for use with a Microsoft Access source database (Windows)” on page 52

Cataloging a target warehouse database for use with a Microsoft Excel data source (Windows)

After you create a target database in DB2 for use with your Microsoft Excel source, you must catalog the database in ODBC.

Procedure:

Cataloging a warehouse target database in ODBC:

1. Click **Start** —> **Settings** —> **Control Panel**.
2. Double-click **ODBC**.
3. Click **System DSN**.
4. Click **Add**.
5. In the **Installed ODBC Drivers** list, select **IBM DB2 ODBC Driver**.
6. Click **OK**.
7. In the **Data Source Name** field, type the database alias.
8. Optional: In the **Description** field, type a description of the database.
9. Click **Select**.
10. Select the path and file name of the database from the list boxes.
11. Click **OK**.
12. Click **Close**.

After you catalog the database in ODBC, define the target and source to the Data Warehouse Center.

Related tasks:

- “Cataloging a target warehouse database for use with a Microsoft Access source database (Windows)” on page 52
- “Defining a warehouse target” on page 84

Defining sources and targets to the Data Warehouse Center that use a Microsoft Excel data source (Windows)

Before you can use a Microsoft Excel source as a warehouse source, you must define the source and target to the Data Warehouse Center.

Setting up non-DB2 warehouse sources

Procedure:

To define the source and target to the Data Warehouse Center:

1. Create a warehouse source for the Microsoft Excel spreadsheet.
2. On the Database page, specify information about the data source.
3. On the Agent Sites page, specify the agent site on which you registered the Microsoft Excel source spreadsheet and the DB2 warehouse database.
4. On the Tables page:
 - a. Expand the **Tables** folder to import the table definition from your Microsoft Excel spreadsheet.
 - b. Click **OK** on the Filter window to list all the tables in your spreadsheet. In the **Available Tables** list, verify that the tables that you created in the database are.
 - c. In the **Available Tables** list, select the table that you want to use.
 - d. Click **>**.

The table moves to the **Selected Tables** list.

5. Create a warehouse target for the DB2 database.
6. Create a step. Give the step the following attributes:
 - It uses one or more source tables from the warehouse source for the Microsoft Excel spreadsheet.
 - It creates a target table in the DB2 warehouse database.
7. Promote the step to test mode.
8. Run the step by right-clicking the step and clicking **Test**.
9. Verify that the data that you created in the Microsoft Excel database is in the warehouse database. Enter the following command in the DB2 Command Line Processor window:

```
select * from prefix.database-name
```

prefix The prefix of the warehouse database (such as IWH).

database-name

The name of the warehouse database.

You will see the data that you entered in the Microsoft Excel database.

Configuring the Red Brick client to access a Red Brick Warehouse database (Windows)

ODBC drivers are used to register the ODBC data source name that the warehouse-supplied Red Brick programs access. You must modify the .odbc.ini file for this purpose.

Prerequisites:

If neither the Red Brick server nor the client is installed, you must install the Red Brick client version 2.20.xC4, including the ODBC driver, on the warehouse agent system where the Red Brick user-defined program is to be run. You can install the Red Brick ODBC Driver from the Red Brick server or the Red Brick client package.

Procedure:

To configure the Red Brick client:

1. Add the Red Brick driver:
 - a. Click **Start** → **Settings** → **Control Panel**.
 - b. Double-click **ODBC**.
 - c. Click **System DSN**.
 - d. Click **Add**.
 - e. Select **Red Brick (32) Driver** from the **Installed ODBC Drivers** list.
 - f. Click **Finish**.
2. In the **Data Source Name** field, type the database DSN name.
 - a. In the **Server(Host:Port)** field, type the name of the server, including host name and port.
 - b. In the **Database Name** field, type the name of the database.
 - c. Optional: In the **Default User Name** field, type a valid user ID.
 - d. Click **OK**.
 - e. Click **Close**.

Configuring non-DB2 warehouse sources (AIX, Linux, Solaris Operating Environment)

To set up the Informix, Sybase Adaptive Server, Oracle, and Microsoft SQL Server sources on AIX, Linux, or the Solaris Operating Environment, see the `odbc.ini` examples in “Setting up connectivity for non-DB2 sources (AIX)” on page 45 and “Setting up connectivity for non-DB2 sources (Solaris Operating Environment, Linux)” on page 47. To see attribute descriptions for each source type, see the DataDirect *DataDirect Connect for ODBC Reference* book published in April of 2003.

For information about configuring the Red Brick client to access a Red Brick warehouse source on AIX, Linux, or the Solaris Operating Environment, see the following topics.

Configuring the Red Brick client to access a Red Brick Warehouse database (AIX, Linux, Solaris Operating Environment)

ODBC drivers are used to register the ODBC data source name that the warehouse-supplied Red Brick programs access. You must modify the `.odbc.ini` file for this purpose.

Prerequisites:

To access a Red Brick Warehouse database, you must install the Red Brick client version 6.20.xC4, including the ODBC driver.

Procedure:

To configure the Red Brick client:

1. Open the `.odbc.ini` file in a text editor.
2. Define your Red Brick Warehouse DSN name in the [ODBC Data Source] section.
3. In the [ODBC] section, provide ODBC environment values for `InstallDir`.
4. For each registered DSN name, add the required information:
 - a. Type the name of the Redbrick driver to be used.

Setting up non-DB2 warehouse sources

- b. Type the server information, including the host name and port.
 - c. Type information about the RB_CONFIG.
 - d. Type the database name.
 - e. Optional: Type the user ID to be used to log in to the database.
 - f. If required, type the password for the user ID.
5. Save and close the file.

Sample .odbc.ini file for the Red Brick ODBC driver

This is a sample .odbc.ini file for setting up ODBC for accessing a Red Brick Warehouse database.

```
#Sample .odbc.ini file for the Red Brick ODBC driver
```

```
#This section is to name your ODBC DSNs
#One entry per DSN
[ODBC Data Source]
RBDSN=Red Brick Driver
```

```
#This section defined the ODBC environment
[ODBC]
InstallDir=<redbrick_dir>
```

```
#This section is to define your ODBC DSNs
#One set of entries per DSN
[RBDSN]
Driver=<redbrick_dir>/lib/rbodbcdrv.so
SERVER=<host>:<port>
RB_CONFIG=<redbrick_dir>
DATABASE=<database_name>
UID=<username>
PWD=<password (optional)>
```

Where <redbrick_dir> is the directory where Red Brick is installed.

Defining a non-DB2 warehouse source in the Data Warehouse Center

You can define non-DB2 data sources as warehouse sources in the Data Warehouse Center.

This procedure is for defining Informix, Sybase, Oracle, and Microsoft SQL Server warehouse sources.

Prerequisites:

Before you define a non-DB2 source to the Data Warehouse Center, you must establish connectivity to the source.

Restrictions:

- If an ODBC data source is created using DB2 Information Integrator and the CREATE NICKNAME statement, the data source will not be available for importing tables in the Data Warehouse Center. To use the data source as a source table, define a non-DB2 warehouse source in the Data Warehouse Center, but do not import a source table. You must create the table manually and ensure that the columns in the warehouse source table map to the columns in the data source. However, if you define the data source as a Client Connect source, the

nickname and its columns can be imported to the Data Warehouse Center. You must verify whether the source columns are appropriate for the source columns selected on the Column Mapping page.

- When you access a warehouse source that was defined using Client Connect with a warehouse agent, the source must be cataloged with the same name, user ID, and password on both the client and warehouse agent systems. If you are using the ODBC version of the warehouse agent, you must also catalog the source as an ODBC source on both the warehouse agent and client sites. Otherwise, actions that require the warehouse agent to access the warehouse source will fail.

Procedure:

To define a non-DB2 database source to the Data Warehouse Center:

1. Open the Warehouse Source notebook in the Data Warehouse Center administrative client.
2. Add information about a warehouse source.
3. Specify an agent site to access a warehouse source.
4. Specify database information for a warehouse source.
5. Import source tables and views into a warehouse source, or create the tables manually.
6. Authorize warehouse groups to access a warehouse source.
7. Specify an agent site to access a warehouse source.
8. Specify how often to retry data extracts.

Related tasks:

- “Defining a warehouse source based on relational databases: Data Warehouse Center help”

Specifying database information for a non-DB2 warehouse source in the Data Warehouse Center

When you define a non-DB2 warehouse source, you must provide information about the data source in the Define Warehouse Source notebook.

Prerequisites:

Before you can define a non-DB2 source to the Data Warehouse Center, you must establish connectivity to the source.

Procedure:

To specify database information for a non-DB2 warehouse source:

1. In the Define Warehouse Source notebook, click the **Data Source** tab.
2. Specify the system data source name in the **Data source name** field.
For non-DB2 sources, the data source name can differ from the database name.
3. If you are using a database or file that has the same name as another database or file on a different workstation, specify the host name of the workstation on which the database or file exists in the **System name** field.
4. In the **User ID** field, type the user ID that will access the database from the agent site workstation.

Setting up non-DB2 warehouse sources

5. In the **Password** field, type the password for the user ID.
6. Type the password again in the **Verify password** field.
7. Optional: Select the **Customize ODBC connect string** check box to enter any additional parameters in the **Additional parameters** field.
8. Go to the Tables and Views page to import the tables and views that you want to use in your source.

Defining warehouse sources for use with a DB2 federated system

A federated system provides several advantages for accessing data. Instead of using ODBC support for non-IBM databases, you can use a federated system to access those databases directly using the native database protocols. You can also use a federated system to write to an Oracle database or other non-IBM databases. With a federated system, you can access and join data from different data sources with a single SQL statement and a single interface. The interface hides the differences between the different IBM and non-IBM databases. A federated system optimizes the SQL statement to enhance performance.

Prerequisites:

- You should be familiar with creating server definitions and nicknames in a DB2 federated system.
- Before you define the warehouse sources, you must create a server definition for each data source server. This definition is stored in the catalog of the federated database.
- You must create nicknames for each data source table that you want to use with the Data Warehouse Center.

Restrictions:

The Data Warehouse Center transformers are not supported with a non-DB2 nickname target database accessed through a federated server.

Procedure:

To define Data Warehouse Center steps that take advantage of the benefits of a federated system, first you define warehouses that use databases accessed through a federated server. Then, you define steps that write to those warehouses.

Define a separate warehouse source definition in the Data Warehouse Center for each source database accessed through a federated server.

You might also need to create a user mapping that maps the DB2 federated server user ID and password to the user ID and password for the source database. The user ID and password that you define in the Data Warehouse Center for the resource is the user ID and password for the corresponding federated server.

Server definitions and nickname tables for warehouse sources accessed through a DB2 federated server

The following example creates a server definition and a nickname for a table:

```
create wrapper sqlnet
create server oracle1 type oracle version 8.1.7 wrapper sqlnet
authorization ihwserve password VWPW options
```



```
(node 'oranode', password 'Y', pushdown 'Y')  
  
create user mapping from USER to server oracle1 authid  
"iwhserve" password "VWPW"  
  
create nickname iwh.oracle_target for ORACLE1.iwhserve.oratar
```

The **create wrapper** statement registers the sqlnet wrapper for the Oracle® data source.

The **create server** statement (extended across multiple lines for readability) defines a data source server called Oracle1, where:

oracle1 The name that you assign to the remote database server. This name is registered in the federated database catalog.

oranode The entry defined in the Oracle TNSNAMES file, which identifies the destination Oracle TCP/IP host and port.

oracle The type of data source.

8.1.7 The version of the data source.

sqlnet The wrapper used to access the database, such as NET8 for Oracle or DBLIB or CTLIB for Sybase.

The DBNAME server option of the **create server** statement is not needed because Oracle allows only one database per database partition. For other data sources, you must specify a database name.

The **create user mapping** statement specifies the user ID that the federated server will use to connect to the remote database (Oracle1). The keyword USER is a DB2® special register that specifies the currently logged-on user. The user connects to the remote database by using the specified user ID and password (iwhserve and VWPW).

The **create nickname** statement specifies a nickname, iwh.oracle_target, for a remote Oracle table, ORACLE1.iwhserve.oratar. After you create the nickname, you must substitute the nickname for the name of the remote table in your SQL statements. For example, if you specify SELECT * FROM iwh.oracle_target, the federated server returns data from the Oracle table iwhserve.oratar.

After you create the mapping and nicknames, you can define the warehouse sources. To define the source tables for each warehouse source, import the federated system nicknames as table definitions.

Defining source tables for federated system warehouse sources

After you create the server mapping and nickname for your table, you can define the warehouse sources.

Procedure:

To define the source tables, import the federated system nicknames as table definitions.

Related concepts:

Setting up non-DB2 warehouse sources

- “Server definitions and nickname tables for warehouse sources accessed through a DB2 federated server” on page 60

Related tasks:

- “Defining warehouse sources for use with a DB2 federated system” on page 60

Chapter 5. Setting up warehouse file sources

Before you can create steps that access your file sources, you must perform the following tasks:

- Determine which file sources you will use in your warehouse.
- Set up connectivity between the file sources and the warehouse agent that you want to use.
- Define the file sources to the Data Warehouse Center as warehouse file sources.

This chapter describes the file sources with which the Data Warehouse Center works and tells you how to set up access to them.

Warehouse agent support for file sources

The following table shows which warehouse agents support which sources.

Table 8. Support for sources on various warehouse agents

Data Source	Windows warehouse agent	AIX or Linux warehouse agent	Solaris Operating Environment warehouse agent	iSeries warehouse agent	zSeries warehouse agent
Local file	✓	✓	✓	✓	✓ ¹
Remote file	✓	✓	✓	✓ ²	✓ ^{1 2}
z/OS file					✓ ¹
VM file	✓				

¹Flat files cannot be accessed as ODBC sources for z/OS or iSeries, but they can be used as a source for warehouse utilities.

²Remote files are supported on iSeries and z/OS only in Copy file using FTP (VWPRCPY) steps. The Copy file using FTP warehouse program is used to copy files on the agent site to or from a remote host.

Defining a file source to the Data Warehouse Center

You can define one or more files in a warehouse source.

Prerequisites:

Set up connectivity for the source.

Restrictions:

You cannot view data in Local File or Remote File warehouse sources before you define the file to Data Warehouse Center.

Procedure:

To define a file source to the Data Warehouse Center:

Setting up warehouse file sources

1. In the Data Warehouse Center, open the Warehouse Source notebook.
2. Specify the properties for your source.
3. Click **OK** to close the Warehouse Source notebook.

Related tasks:

- “Defining a warehouse source based on files: Data Warehouse Center help”

Accessing remote data files

You can access data files on a remote workstation by the automatic transfer of the file through FTP from the remote machine to the agent site. Only an SQL Insert step can use a remote source file. The SQL Update step supports only a table source. When you promote a step that uses remote files to test mode, the files will be transferred just as if the step is in production mode. If the file is large, promotion to test mode or production mode can take a long time, and the agent site can run out of space. In general, remote files should be limited to 20 megabytes or smaller to avoid problems.

One way to avoid running out of space is to place a dummy file on the remote workstation during testing. Using the Copy file using FTP warehouse program instead of automatic FTP is another way to avoid running out of space.

When you promote the step that uses this source to test mode, the Data Warehouse Center transfers the file to a temporary file on the agent site.

Prerequisites:

If you plan to use the FTP commands GET or PUT to transfer files from a z/OS host to another remote host, you must create a JCL template for your z/OS system. The JCL templates are stored in the directory defined to by the environment variable VWS_TEMPLATES.

To create a JCL template for your z/OS system:

1. Copy the template ftp.jcl to hostname.ftp.jcl, where hostname is the z/OS hostname.
2. Add the valid JCL account information for the z/OS system by updating the job card of the JCL in this template. Contact your system programmer for the valid JCL account information for the z/OS system that the JCL is running on.

Procedure:

To access a remote data file using automatic FTP:

1. Store the file on an FTP site.
2. Declare the file with a warehouse source type of Remote File.
3. Specify the system name, user ID, and password in the Advanced window that you open from the Files notebook page of the Warehouse Source notebook.
4. Connect the file as a source to an SQL Insert step.

Related tasks:

- “Accessing data files with the Copy file using FTP warehouse program” on page 65
- “Manipulating files using FTP or the Submit JCL jobstream warehouse program” on page 133

Accessing data files with the Copy file using FTP warehouse program

You can use the Copy file using FTP warehouse program to access data files on a remote workstation. The Copy file using FTP warehouse program is an explicit way to transfer (GET) files from a remote machine to the agent site, or to transfer (PUT) files from the agent site to a remote machine. When the file size is larger than 20 megabytes, use the Copy file using FTP warehouse program in preference to the automatic FTP GET process. When you use the Copy file using FTP warehouse program, define the source and target files as local files.

Procedure:

To use the Copy file using FTP warehouse program to access a file:

1. Declare the file with a warehouse source and warehouse target type of Local File.
2. Define a step that uses the Copy file using FTP warehouse program to copy the file from the remote machine to the agent site. You must specify the machine name, user ID, and password for the remote machine as parameters in this step.
3. Run the Copy file using FTP warehouse program step to transfer the file to the agent site.
4. Define the target file from the Copy file using FTP program step as the source of the SQL Insert step. The step will access the target file as a local file. This file is the output file of the previous step.

The same agent runs both of these steps.

Related tasks:

- “Copying files to and from a remote host: Data Warehouse Center help”

Setting up connectivity for file sources (Windows NT, Windows 2000, Windows XP)

The following topics explain how to set up connectivity for file sources on Windows.

Setting up connectivity for a z/OS or VM file source (Windows NT, Windows 2000, Windows XP)

When you set up a data warehouse to use a z/OS or VM file source, you must set up connectivity to it.

Prerequisites:

Database access program

FTP or NFS

Source/agent connection

TCP/IP (FTP or NFS)

Procedure:

To connect to a z/OS or VM file source, establish a link from the agent site to the host.

Setting up connectivity for a local file source (Windows NT, Windows 2000, Windows XP)

When you set up a data warehouse to use a local file source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

DataDirect Driver Manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Procedure:

To connect to a local file source, supply a pre-access command, a post-access command, or both as required.

Requirements for accessing a remote file from a file server (Windows)

You can use data files as a source file for a step. If the file is not on the agent site, but is accessed through a Windows® file server, you must meet the following requirements. The requirements for accessing a remote file on a LAN server are similar to these requirements.

The agent site must have a user ID and password that is authorized to access the file. The agent site must contain a .bat file that performs the NET USE command. The file must contain at least the following lines:

```
NET USE drive: /DELETE
```

drive The drive letter that represents the shared drive on the agent site.

hostname

The TCP/IP hostname of the remote workstation.

sharedDrive

The drive on the remote workstation that contains the file.

password

The password that is required to access the shared drive.

userid

The user ID that is required to access the shared drive.

The first command in the file releases the drive letter if it is in use. The second command establishes the connection.

When you define the warehouse source for the file, you must specify the .bat file in the **Pre-Access Command** field in the Advanced window that you open from the Files notebook page of the Warehouse Source notebook.

When you do this, specify the .bat file in the **Post-Access Command** field in the Advanced window.

Setting up connectivity for a remote file source (Windows NT, Windows 2000, Windows XP)

When you set up a data warehouse to use a remote file source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

ODBC

Client enabler program

DataDirect Driver Manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Procedure:

To set up connectivity for a remote file source:

1. Obtain the FTP server or mount string.
2. Identify the user ID and password with access to the source files.

Setting up connectivity for file sources (AIX)

The following topics explain how to set up connectivity for file sources on AIX.

Setting up connectivity to a z/OS or VM file source (AIX)

When you set up a data warehouse to use a z/OS, or VM file source, you must set up connectivity to it. The z/OS agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Prerequisites:

Database access programs

FTP or NFS

Source/agent connection

TCP/IP (FTP or NFS)

Client enabler program

DataDirect Driver Manager and text file driver

Restriction:

The zSeries warehouse agent cannot connect to file sources. You read file sources through standard commands.

Procedure:

Setting up warehouse file sources

To set up connectivity for a z/OS or VM file source, establish a TCP/IP communication from the agent site to the host.

Setting up connectivity for a local file source (AIX)

When you set up a data warehouse to use a local file source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

TCP/IP

Client enabler program

DataDirect Driver Manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Procedure:

If required, supply a pre-access command, a post-access command, or both to connect.

Related concepts:

- “Requirements for accessing a remote file from a file server (Windows)” on page 66

Related tasks:

- “Accessing remote data files” on page 64

Setting up connectivity for a remote file source (AIX)

When you set up a data warehouse to use a remote file source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

TCP/IP

Client enabler program

DataDirect Driver Manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Procedure:

To set up connectivity for a remote file source:

1. Obtain the FTP server or mount string.
2. Identify the user ID and password with access to the source files.

Setting up connectivity for file sources (Solaris Operating Environment, Linux)

The following topics explain how to set up connectivity for file sources on the Solaris Operating Environment and Linux.

Setting up connectivity for a z/OS or VM file source (Solaris Operating Environment, Linux)

When you set up a data warehouse to use a z/OS or VM file source, you must set up connectivity to it.

Prerequisites:**Database access programs**

FTP or NFS

Source/agent connection

TCP/IP (FTP or NFS)

Restriction:

The zSeries warehouse agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Procedure:

To set up connectivity for a z/OS or VM file source, establish a TCP/IP communication from the agent site to the host.

Setting up connectivity for a local file source (Solaris Operating Environment, Linux)

When you set up a data warehouse to use a local file source, you must set up connectivity to it.

Prerequisites:**Source/agent connection**

TCP/IP

Client enabler program

DataDirect Driver Manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Procedure:

To set up connectivity for a local file source, supply a pre-access command, a post-access command, or both as required.

Setting up connectivity for a remote file source (Solaris Operating Environment, Linux)

When you set up a data warehouse to use a remote file source, you must set up connectivity to it.

Prerequisites:

Source/agent connection

TCP/IP

Client enabler program

DataDirect Driver Manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. You access file sources through standard open, read, and close commands.

Procedure:

To set up connectivity for a remote file source:

1. Obtain the FTP server or mount string.
2. Identify the user ID and password with access to the source files.

Example .odbc.ini file entry for a warehouse file source (AIX, Solaris Operating Environment, Linux)

The following is an example of an .odbc.ini file entry for a warehouse file source.

```
[IWH_TEXT]
#for aix
Driver=/usr/opt/db2_08_01/odbc/lib/ibtxt16.so
#for linux/solaris
#Driver=/opt/IBM/db2/V8.1/odbc/lib/ibtxt16.so
Description=Text driver
```

Related tasks:

- “Setting up connectivity for a z/OS or VM file source (Windows NT, Windows 2000, Windows XP)” on page 65
- “Setting up connectivity for a local file source (Windows NT, Windows 2000, Windows XP)” on page 66
- “Setting up connectivity for a remote file source (Windows NT, Windows 2000, Windows XP)” on page 67
- “Setting up connectivity to a z/OS or VM file source (AIX)” on page 67
- “Setting up connectivity for a local file source (AIX)” on page 68
- “Setting up connectivity for a remote file source (AIX)” on page 68
- “Setting up connectivity for a z/OS or VM file source (Solaris Operating Environment, Linux)” on page 69
- “Setting up connectivity for a local file source (Solaris Operating Environment, Linux)” on page 69
- “Setting up connectivity for a remote file source (Solaris Operating Environment, Linux)” on page 70
- “Defining a file source to the Data Warehouse Center” on page 63

Network File System protocol

You can access files from an agent site by using the Network File System (NFS) protocol of TCP/IP. When you use NFS, you must provide a user ID on the NFS command. You must specify the access commands in the **Pre-Access Command** field in the Advanced window that you open from the Files notebook page of the Warehouse Source notebook.

If the agent site does not have NFS installed, use the NET USE command to access NFS.

To use a source data file, you also must register the file with ODBC as a system DSN of IWH_TEXT. Use an appropriate driver, such as DataWHSE 3.7 32-bit Textfile (*.*). The system DSN is defined automatically when the Data Warehouse Center is installed on Windows[®] NT, Windows 2000, or Windows XP. On a UNIX[®] system, you must enter the correct definition for IWH_TEXT in the .odbc.ini file.

For example:

```
[Data Sources]
IWH_TEXT= Flat file db
.
.
.
.
.
[IWH_TEXT]
Driver=/usr/opt/db2_08_01/odbc/lib/ibtxt16.so
Description=Text driver
```

Related tasks:

- “Setting up connectivity for a local file source (Windows NT, Windows 2000, Windows XP)” on page 66
- “Setting up connectivity for a remote file source (Windows NT, Windows 2000, Windows XP)” on page 67
- “Setting up connectivity for a local file source (AIX)” on page 68
- “Setting up connectivity for a remote file source (AIX)” on page 68
- “Setting up connectivity for a local file source (Solaris Operating Environment, Linux)” on page 69
- “Setting up connectivity for a remote file source (Solaris Operating Environment, Linux)” on page 70
- “Defining a file source to the Data Warehouse Center” on page 63

Setting up warehouse file sources

Chapter 6. Setting up access to a warehouse

When you create a warehouse, you must set up access to it. You can use any application that can access DB2 data to access your warehouse database. This chapter will show you how to set up access to the following warehouse databases:

- DB2 Universal Database
- DB2 for iSeries
- DB2 for z/OS
- Warehouse databases with multiple partitions
- Warehouse databases that are accessed by DB2 Relational Connect

| This chapter also explains how to create a Data Warehouse Center status table,
| how to create and remove target tables in a remote database, and how to move
| target tables to a remote database.

Setting up a DB2 Universal Database warehouse

This section explains how to set up a DB2 Universal Database warehouse. To set up a DB2 Universal Database warehouse, you need to define privileges and set up connectivity.

Defining privileges for DB2 Universal Database warehouses

| You define privileges so that specified users have the right to access the warehouse
| and perform tasks. To set up a warehouse, you must have a database on the target
| system and a user ID with the authorization to define privileges.

Procedure:

To define privileges for DB2 Universal Database warehouses:

1. Set up a user ID with the following privileges for the database:
 - CREATETAB (database-level privilege)
 - BINDADD (database-level privilege)
 - CONNECT (database-level privilege)
2. Assign an explicit SELECT privilege to the following system tables:
 - SYSIBM.SYSTABLES
 - SYSIBM.SYSCOLUMNS
 - SYSIBM.SYSDBAUTH
 - SYSIBM.SYSTABAUTH
 - SYSIBM.SYSINDEXES
 - SYSIBM.SYSRELS
 - SYSIBM.SYSTABCONST

After you define privileges for DB2 Universal Database warehouses, establish connectivity to the warehouse.

Connecting to DB2 Universal Database and DB2 Enterprise Server Edition warehouses

After you define the required privileges for the DB2 Universal Database or DB2 Enterprise Server Edition warehouse, you can establish connectivity.

Procedure:

To establish connectivity, perform the following tasks at the agent site:

1. Set up communications to the target system, if the database is remote.
2. Catalog the node of the target system if the database is remote.
3. Catalog the database.
4. Register the database as an ODBC system DSN if you are using the Windows NT, Windows 2000, or Windows XP warehouse agent, or the version of the AIX, Linux, z/OS, or Solaris Operating Environment warehouse agent that uses ODBC. If you are using the AIX, Solaris Operating Environment, or Linux warehouse agent that uses the CLI interface, catalog the database using DB2 catalog utilities.
5. Bind database utilities and ODBC(CLI) to the target database. Perform the bind only once from each type of client.

Related tasks:

- “Defining privileges for DB2 Universal Database warehouses” on page 73

Setting up a DB2 for iSeries warehouse

You can use DB2 Connect or Client Access/400 to access a DB2 for iSeries warehouse. You need to define privileges, establish connectivity, and verify that the servers are running.

Defining privileges to DB2 for iSeries warehouses

Before you can access a DB2 for iSeries warehouse, you must define privileges for it.

Prerequisites:

The system administrator of the target system must set up a user ID with CHANGE authority or higher on the NULLID collection.

Procedure:

To define privileges for an iSeries warehouse:

1. Assign an explicit SELECT privilege for the following system tables:
 - SYSIBM.SYSTABLES
 - SYSIBM.SYSCOLUMNS
 - SYSIBM.SYSINDEXES
 - SYSIBM.SYSREFCST
 - SYSIBM.SYSCST
2. Assign the ALLOBJ privilege to the user ID to create iSeries collections.
3. Set up a DB2 Connect gateway site.

Setting up a DB2 Connect gateway site (iSeries)

After you define the required privileges for accessing the DB2 for iSeries warehouse, you can set up a gateway site.

Procedure:

To set up a DB2 Connect gateway site, perform the following tasks at the gateway site:

1. Install DB2 Connect.
2. Configure your DB2 Connect system to communicate with the target database.
3. Update the DB2 node directory, system database directory, and DCS directory with the values for the target database.

After you set up the DB2 Connect gateway site, you can establish connectivity to the DB2 for iSeries warehouse.

Connecting to a DB2 for iSeries warehouse with DB2 Connect

You can establish connectivity to the DB2 for iSeries warehouse.

Prerequisites:

- Define privileges to the DB2 for iSeries warehouse
- Set up the DB2 Connect gateway site

Procedure:

To establish connectivity, perform the following tasks at the agent site:

1. Set up communications to the DB2 Connect site.
2. Catalog the node of the DB2 Connect site.
3. Catalog the database.
4. Register the database as a system DSN by using the ODBC Administrator.
5. Bind the DB2 Connect utilities to the DRDA server. Perform the bind only once from each type of client.

Related tasks:

- “Setting up a DB2 Connect gateway site (iSeries)” on page 75
- “Defining privileges to DB2 for iSeries warehouses” on page 74

Setting up a DB2 for z/OS warehouse

This section describes the steps that are necessary for setting up a DB2 for z/OS warehouse. You need to define privileges and establish connectivity before you can define a warehouse.

Setting up a DB2 Universal Database for z/OS warehouse

By default, the Data Warehouse Center creates warehouse target tables in a format that is supported by the DB2 Universal Database products. You also can customize steps to work with target tables in DB2 for z/OS.

Procedure:

Setting up access to a warehouse

To set up a DB2 for z/OS warehouse:

1. Define privileges for the DB2 for z/OS warehouse.
2. Establish connectivity to the DB2 for z/OS warehouse database.
3. Define the warehouse to the Data Warehouse Center.

Defining privileges for a DB2 for z/OS warehouse

You must define privileges for a DB2 for z/OS warehouse before you can connect to it.

Prerequisites:

Before setting up a warehouse, you must create a database on the target system.

Procedure:

To define privileges for a DB2 for z/OS warehouse:

1. Create a database and table space to use as the warehouse.
2. Set up a user ID with DBADM privileges for the database.

Connecting to a DB2 for z/OS warehouse database

Before you can define a DB2 for z/OS warehouse to the Data Warehouse Center, you must establish connectivity to it.

Prerequisites:

Define privileges for the DB2 for z/OS warehouse.

Procedure:

To set up connectivity to the DB2 for z/OS warehouse:

1. Set up access to DB2 for z/OS.
2. Connect to the warehouse database.

Related tasks:

- “Defining privileges for a DB2 for z/OS warehouse” on page 76
- “Defining a DB2 for z/OS warehouse to the Data Warehouse Center” on page 76

Defining a DB2 for z/OS warehouse to the Data Warehouse Center

When you define a target table for a DB2 for z/OS warehouse, you must specify a table space in which to create the table. If you do not specify a table space, DB2 for z/OS creates the table in the default DB2 database that is defined for the given subsystem.

Procedure:

To define the DB2 for z/OS warehouse to the Data Warehouse Center:

1. Define a warehouse.
2. Define or generate a target table.

3. Right-click the target table, and click **Properties** to open the Properties notebook for the table.
4. In the **Table space** field, specify the table space in which to create the table.
5. Verify that the **Grant to public** check box is clear.
6. Click **OK** to save your changes and close the Table notebook.

When you promote the step to test mode, if you specified that the Data Warehouse Center is to create the target table, the Data Warehouse Center creates the target table in the DB2 for z/OS.

Related tasks:

- “Defining privileges for a DB2 for z/OS warehouse” on page 76
- “Connecting to a DB2 for z/OS warehouse database” on page 76
- “Setting up a DB2 Universal Database for z/OS warehouse” on page 75

Setting up a warehouse on a database with multiple partitions

By default, the Data Warehouse Center creates warehouse target tables in a format that is supported by a database with a single partition. You can design steps to work with target tables in a database with multiple partitions.

Defining privileges to a database with multiple partitions

Before you can establish connectivity to a warehouse in a database with multiple partitions, you must define privileges to that database.

Prerequisites:

You must create a database on the target system.

Procedure:

To define privileges to a database with multiple partitions:

1. Create a database, database partition group, and table space to use as the warehouse.
2. Give a user ID BINDADD, CONNECT, and CREATETAB privileges for the database.
3. Assign an explicit SELECT privilege to the following system tables:
 - SYSIBM.SYSTABLES
 - SYSIBM.SYSCOLUMNS
 - SYSIBM.SYSDBAUTH
 - SYSIBM.SYSTABAUTH
 - SYSIBM.SYSINDEXES
 - SYSIBM.SYSRELS
 - SYSIBM.SYSTABCONST

After you define privileges to the database, you can establish connectivity.

Defining a database with multiple partitions to the Data Warehouse Center

After setting up access to the database with multiple partitions, you need to define the database.

Prerequisites:

- Define privileges to the database with multiple partitions.
- Establish connectivity to the database with multiple partitions.

Procedure:

To define the database with multiple partitions to the Data Warehouse Center:

1. Define a warehouse target.
2. Define or generate a target table.
3. Right-click the target table, and click **Properties** to open the Table notebook.
4. Click **Show SQL** to open the Show SQL window.
5. Click **Edit**.

A confirmation window opens. Click **Yes** to edit the SQL.

6. Add the following clause to the end of the CREATE statement:

```
IN tablespace-name PARTITIONING KEY (column-name) USING HASHING
```

```
tablespace-name
```

The name of the table space that contains the database.

```
column-name
```

The name of the column to use for partitioning.

7. Click **OK**.
The Edit SQL window closes.
8. Click **OK** to save your changes and close the Table notebook.

When you promote the step to test mode, the Data Warehouse Center creates the target table in the database with multiple partitions.

Creating the Data Warehouse Center status table

Users can use the BVBESTATUS table to join tables by matching their timestamps or query editions by date range rather than by edition number.

For example, the edition number 1010 might not have any meaning to a user, but the dates on which the data was extracted might have meaning. You can create a simple view on the target table to allow users to query the data by the date on which it was extracted.

You must manually create the status table.

Procedure:

To create the Data Warehouse Center status table, if your target warehouse is in DB2 on a Windows system, DB2 for iSeries, or DB2 on a UNIX-based system, enter the following DB2 command:

```
CREATE TABLE IWH.BVBESTATUS (BVNAME VARCHAR(80) NOT NULL,  
RUN_ID INT NOT NULL, UPDATIME CHAR(26) NOT NULL);
```

If your target warehouse is in DB2 for z/OS, you might need to identify the names of the database and table space where the status table is to be created. Add the following clause to the end of the CREATE TABLE statement:

```
IN database-name.tablespace-name
```

database-name

The name of the database in which you want to create the BVBESTATUS table.

tablespace-name

The name of the table space in which you want to create the BVBESTATUS table.

If your target warehouse is in a DB2 partitioned database, you might need to identify the name of the table space where the status table is to be created and the partitioning key for the database. Add the following clause to the end of the CREATE TABLE statement:

```
IN tablespace-name partitioning-key (RUN_ID)  
USING HASHING
```

tablespace-name

The name of the table space in which you want to create the BVBESTATUS table.

partitioning-key

The name of the column that specifies in which partition a particular row of data is stored.

Related tasks:

- “Defining a warehouse for use with a federated system” on page 79

Defining a warehouse for use with a federated system

A federated system provides several advantages for accessing data for steps. Instead of using ODBC support for non-DB2 databases, you can use a federated system to access those databases directly using the native database protocols. You can also use a federated system to write to non-DB2 databases and DB2 databases. With a federated system, you can access and join data from different data sources with a single SQL statement and a single interface. The interface hides the differences between the different IBM and non-DB2 databases. A federated system optimizes the SQL statement to enhance performance.

You can define Data Warehouse Center steps that take advantage of a federation. First, you define warehouses that use databases accessed through a DB2 federated server. Then you define steps that write to those warehouses.

You can use SQL to extract data from the source database and write the data to the target database. When the Data Warehouse Center generates the SQL to extract data from the source database and write data to the target database, the Data Warehouse Center generates an INSERT FROM SELECT statement because the federated database is both the source and target database. The DB2 query optimizer then optimizes the query for the federated system target databases (such as Oracle and Sybase).

To access the warehouse with a different Data Warehouse Center agent, change the name of the database, the user ID, and the password.

Setting up access to a warehouse

The BVBESTATUS table contains timestamps for the step editions in the warehouse database.

If you create the BVBESTATUS table in the remote databases, updates to the table will be in the same commit scope as the remote databases. You must have a different federated database for each remote database, because the Data Warehouse Center requires that the name of the table be BVBESTATUS. One nickname cannot represent multiple tables in different databases.

To create the BVBESTATUS table, use the CREATE TABLE statement. For example, to create the table in an Oracle database, issue the following command:

```
CREATE TABLE BVBESTATUS (BVNAME, VARCHAR2(80) NOT NULL,  
                          RUN_ID NUMBER(10) NOT NULL,  
                          UPDATIME CHAR(26) NOT NULL)
```

After you create the table, create a nickname for the IWH.BVBESTATUS table on the federated server.

Prerequisites:

- You should be familiar with the basic components of a warehouse and configuring access to data sources.
- You must create a BVBESTATUS table in the federated database or in each remote database.
- If you create the BVBESTATUS table in the federated database, you must use two-phase commit to keep the table synchronized with the remote warehouse databases.

Procedure:

To define a warehouse for use with a federated system:

1. Create a separate logical warehouse definition in the Data Warehouse Center for each federated target database. For example, define one warehouse for each Oracle database, another warehouse for each Sybase database, and so on.
2. Specify the same federated database as the database for each logical warehouse.

Related tasks:

- “Setting up a DB2 Universal Database for z/OS warehouse” on page 75

Creating target tables in a remote database

With a federated system, using DB2 Information Integrator, the Data Warehouse Center can create tables in a remote database, such as Oracle.

Procedure:

To create the target table:

1. Create a step with a target table in a federated database.
2. In the Table notebook, type the server mapping name in the **Table space** field.
3. Determine what you need to do because the default table name qualifier for the Data Warehouse Center is IWH:
 - Request a user ID named IWH.
 - Change the table name qualifier to match your user ID.

- Request that your user ID be granted the privilege to create any table, insert into any table, update any table, and drop any table.
- Request that your user ID be granted DBA authority.

If your user ID for the target database has the privilege to create a table with a qualifier that is different from your user ID, you can promote the step to test mode.

4. Promote the step to test mode.
5. Run the step to verify that the correct data is written to the target table.
6. Promote the step to production mode.

In DB2 Universal Database Version 8, DB2 Information Integrator replaces and enhances the functions provided by DB2 Relational Connect. If you are using DB2 Relational Connect with DB2 Version 7 on Windows NT or Windows 2000, or UNIX and Version 8.1.2 or later you might get an error indicating a bind problem. For example, when using a DB2 Relational Connect source with a Data Warehouse Center Version 7 agent, you might get the following error:

```
DWC07356E An agent's processing of a command of type
"importTableNames" failed for edition "0" of step "?".
SQL0001N Binding or precompilation did not complete
successfully.
SQL0001N Package "NULLID.SQLL6D05" was not found.
SQLSTATE=51002 RC = 7356 RC2 = 8600
```

To correct the problem, add the following lines to the db2cli.ini file:

```
[COMMON]
DYNAMIC=1
```

On UNIX systems, the db2cli.ini file is located in the .../sqllib/cfg directory. On Windows NT, Windows 2000, Windows XP the db2cli.ini file is located in the .../sqllib directory.

Related tasks:

- “Defining a warehouse for use with a federated system” on page 79
- “Moving a target table from a federated database to a remote database” on page 81
- “Updating a table in a remote database” on page 82

Moving a target table from a federated database to a remote database

You can create and test a step in a federated database, and then move it to a remote database. For example, you want to populate an Oracle table with data from a DB2 table. First, you create an SQL step that defines the target table as a warehouse-created table on the federated server. When the step is promoted to test mode, the Data Warehouse Center creates the table with the correct column definitions. You can test the step to make sure that it populates the target table correctly. Then you can use the target table definition to create the table on Oracle with the correct column definitions by using a tool such as Oracle SQL*Plus. Next you drop the table from the federated server and create a nickname for the Oracle table. At this point, you can promote the warehouse step to production mode and run the step to populate the Oracle table.

Prerequisites:

1. Create a step with a target table in a federated database.

Setting up access to a warehouse

-
2. Promote the step to test mode.
3. Check that the data types of the federated tables and the remote tables are compatible.

Procedure:

To move a target table in a federated database to a remote database:

1. Run the step to verify that the connections to the source databases are working and that the correct data is written to the target table.
2. Move the table to a remote database by creating the table in the remote database and dropping the federated table. (You can also use a modeling or data dictionary tool.)
3. Create a nickname for the remote table in the federated database catalog. The nickname must match the name of the target table for the step in the Data Warehouse Center.
4. Run the step again to test that the data moves through the federated server to the target correctly.
5. Promote the step to production mode.

Updating a table in a remote database

You can use the Data Warehouse Center to update an existing table in a remote database. Use this option when data already exists or when you are using another tool, such as a modeling tool, to create the warehouse schema.

Procedure:

To update a table in a remote database:

1. Create a nickname for the target table in the federated database catalog.
2. Import the nickname definition into a warehouse by selecting the table in the **Tables** list in the Define Warehouse Target notebook.
In the Properties notebook for the table, the Columns page displays the imported table definition.
3. Create a step in the warehouse, selecting the imported table as the target table for the step. If there is a **Data Warehouse Center Created Table** check box on the Processing Options page of the step notebook, verify that it is clear when you promote the step.
4. Promote the step to test mode.
5. Run the step to test access to the target tables.
6. Promote the steps to production mode.

Chapter 7. Setting up a warehouse target

You can create a warehouse target to hold the data that is moved and transformed from your warehouse source.

Warehouse targets

Any warehouse user can define a warehouse target, but only users who belong to a warehouse group with access to the warehouse target can change the warehouse target. Table names can be case sensitive, or have other restrictions, for some warehouse target types.

If the warehouse target has more than one agent site selected, the warehouse server uses the agent site with the name that sorts first (depending on the user's locale) for the import. For example, your warehouse target has three agent sites selected: Default Agent, AIX[®] Agent, and zSeries[®] Agent. The warehouse server uses the agent site named AIX Agent for the import.

If you change the properties of a source or target, the change that you made is automatically saved. If you make any other change, such as adding a step, you must explicitly save the change to make the change permanent. To save the change, click **Process** → **Save**.

When defining a warehouse target for z/OS[™], the option to create Data Warehouse Center transformers in the target database is not available, because z/OS does not support scripts.

Ordinary identifier support in the Data Warehouse Center

The Data Warehouse Center supports target tables that use ordinary SQL identifiers. An ordinary identifier:

- Must start with a letter.
- Can include uppercase letters, numbers, and underscores.
- Cannot be a reserved word.

If a table has a lowercase letter as part of its ordinary identifier, the Data Warehouse Center stores the lowercase letter as an uppercase letter.

Delimited identifier support in the Data Warehouse Center

The Data Warehouse Center supports target resource tables that use delimited identifiers. A delimited identifier:

- Is enclosed within quotation marks.
- Can include uppercase and lowercase letters, numbers, underscores, and spaces.
- Can contain a double quotation mark, represented by two consecutive quotation marks: "".

Edition columns

When you set the number of editions and specify the edition column in the warehouse target properties, set the number of editions to 0 if you want the Data Warehouse Center to append data each time the process runs.

Setting up a warehouse target

The system-generated column is treated like any other user-defined column. System-generated columns are not removed automatically when the number of editions is set to 0, or when a different column is selected in the Edition column list. If the column is not selected as the edition column, you can remove it manually from the Columns page in the Properties notebook for the table.

The System-generated option is removed from the Edition column list when an edition column is already specified. To return the System-generated option to the list, first set the number of editions to 0, then set the number of editions to a value greater than 0. If you create more than one system-generated edition column, the columns are given unique names. For example, the first column is called RUN_ID, the second column is called RUN_ID1, the third column is called RUN_ID2, and so on.

If you specified a column as an edition column, and you then select a different column from the list, the new column is now the edition column. The old edition column remains in the table, but is no longer specified as the edition column. You must manually remove the old edition column if you do not want it in the table.

Supported warehouse targets

The following table shows the versions and release levels of supported DB2® warehouse targets.

Target	Version
DB2 Universal Database™ for Windows NT	7 through 8.1
DB2 Universal Database Enterprise Server Edition	8.1
DB2 Universal Database Enterprise-Extended Edition	6 through 7.x
DB2 Universal Database for iSeries	5.2 and later
DB2 Universal Database for AIX	7 through 8.1
DB2 Universal Database for Solaris Operating Environment	7 through 8.1
DB2 Universal Database for Linux	8.1
DB2 Universal Database for z/OS™	6 and later
DB2 Information Integrator	8.1
DB2 for VM	7.3
DB2 for VSE	7.3

In addition to the target databases supported above, the Client Access/400 Version 3.1.2 tool is also supported.

Defining a warehouse target

After you define the sources for your warehouse as warehouse sources, define the warehouse target that will contain the data.

Restrictions:

If a data source is created using DB2 Information Integrator and the Create Nickname statement, and is defined as a warehouse target through ODBC, the warehouse agent will not be able to import the tables. To use the data source as a

target table, define a warehouse target in the Data Warehouse Center, but do not import a target table. Create the table manually and ensure that the columns in the warehouse target table map to the columns in the data source. However, if you define the warehouse target as a Client Connect target nickname, then its columns can be imported to the Data Warehouse Center. You must verify whether the target columns are appropriate for the source columns selected on the Column Mapping page.

Procedure:

To define a warehouse target:

1. Verify the version and release level of your warehouse target to ensure that it is supported.
2. Define your warehouse target.
3. Optional: Define a primary key.
4. Optional: Define foreign keys.

Related concepts:

- “DB2 warehouse sources” on page 37

Related tasks:

- “Defining DB2 warehouse sources” on page 38

Warehouse primary and foreign keys

Warehouse primary keys and warehouse foreign keys are definitions that you create to describe the underlying database table keys.

A key for a table is one or more columns that uniquely identify the table. The warehouse primary key for a table is one of the possible keys for the table that you define as the key to use.

You can define foreign keys for a warehouse source table, warehouse source view, or warehouse target table. The Data Warehouse Center uses foreign keys only in the join process. The Data Warehouse Center does not commit foreign keys that you define to the underlying database.

Before you define foreign keys, you must know the name and schema of the parent table to which the foreign keys correspond.

You can define foreign keys when the step is in development or test mode. If you specify a primary key for a table on z/OS™, you must define a unique index for the table after it is created.

Related concepts:

- “Star joins” on page 116

Related tasks:

- “Defining a warehouse primary key: Data Warehouse Center help”
- “Defining foreign keys: Data Warehouse Center help”

Running DB2 Warehouse Manager on a 64-bit DB2 instance

You can run the DB2 Warehouse Manager Standard Edition on a 64-bit operating system under a 32-bit DB2 instance. In addition, you can use the DB2 Warehouse Manager to move data on a 64-bit DB2 instance, but only if you run all of the warehouse components, such as the warehouse server, warehouse control database, and warehouse agent, on a 32-bit DB2 instance.

Prerequisites:

- The warehouse server, warehouse control database, and the warehouse agent must reside on a 32-bit DB2 instance.
- You must define the 64-bit warehouse source and the 64-bit warehouse target as remote instances.

Procedure:

To run the DB2 Warehouse Manager on the same workstation as a 64-bit warehouse source and a 64-bit warehouse target:

1. Define a 32-bit DB2 instance by entering the following command at a DB2 command line:
`db2icrt`
2. Verify that the IWH.environment file points to the 32-bit DB2 instance that you defined. Make sure that the warehouse server, warehouse agent, and warehouse control database reside on this 32-bit DB2 instance.
3. In the 32-bit instance that contains the warehouse components, catalog the 64-bit warehouse source database and the 64-bit warehouse target database by entering the following command from a DB2 command line:
`CATALOG TCP/IP NODE node-name SERVER service-name REMOTE_INSTANCE instance-name
CATALOG DB 64bitDB AT NODE node-name`

Related tasks:

- “Defining an agent site” on page 18
- “Defining a warehouse target” on page 84
- “Defining DB2 warehouse sources” on page 38

Chapter 8. Defining and running processes

This chapter describes how to define and run processes in the Data Warehouse Center

Warehouse processes

After you define a warehouse, you need to populate the warehouse with useful information. To do this, you need to understand what the users need, what source data is available, and how the Data Warehouse Center can transform the source data into information.

To identify and group processes that relate to a logical area of the business, you first define a subject area.

For example, if you are building a warehouse of sales and marketing data, you define a Sales subject area and a Marketing subject area. You then add the processes that relate to sales under the Sales subject area. Similarly, you add the definitions that relate to the marketing data under the Marketing subject area.

To define how data is to be moved and transformed for the data warehouse, you define a process, which contains a series of steps in the transformation and movement process, within the subject area.

Within the process, you define data transformation steps that specify how the data is to be transformed from its source format to its target format. Each step defines a transformation of data from a source format to a target format, by including the following specifications:

- One or more source tables, views, or files from which the Data Warehouse Center is to extract data.

You must define these sources as part of a warehouse source before you can use the source tables in a step.

- A target table to which the Data Warehouse Center is to write data.

You can specify that the Data Warehouse Center is to create the table in a warehouse database, according to your specifications in the step, or you can specify that the Data Warehouse Center is to update an existing table.

- How the data is to be transformed:

- By issuing an SQL statement that specifies the data to extract and how to transform the data to its target format.

For example, the SQL statement can select data from multiple source tables, join the tables, and write the joined data to a target table.

- By running a warehouse program or transformer.

For example, you might want to use the DB2[®] bulk load and unload utilities to transfer data to your warehouse. Or you might want to use the Clean transformer to clean your data. You can also define an external program to the Data Warehouse Center as a user-defined program.

- By running a user-defined program.

Defining the transformation and movement of data within the Data Warehouse Center

After you define warehouse sources and targets, you can move and transform the data within the Data Warehouse Center.

Procedure:

To define data transformation and movement:

1. Define a subject area.
2. Define a process.
3. Define steps.
4. Schedule the steps or the process.
5. Monitor the progress of your steps.

For more information about performing these steps, see the related tasks. The related topics provide links to information about how to complete these steps based on the different programs and steps that you want to define.

Related concepts:

- “Warehouse processes” on page 87

Related tasks:

- “Selecting and running new steps: Data Warehouse Center help”
- “Steps and tasks: Data Warehouse Center help”

Warehouse steps

You need to create steps that define how the source data is to be moved and transformed into the target data. The following list describes the main step types.

SQL steps

The Data Warehouse Center provides two kinds of SQL steps. A Select and Insert SQL step uses an SQL SELECT statement to extract data from a warehouse source and generates an INSERT statement to insert the data into the warehouse target table. A Select and Update SQL step uses an SQL UPDATE statement to update data in the warehouse target table.

Warehouse program steps

Warehouse program steps run predefined programs and utilities. There are several types of warehouse program steps:

- File programs
- OLAP programs
- Warehouse deprecated programs
- Programs for specific operating systems

The warehouse programs for a particular operating system are packaged with the agent for that operating system. You install the warehouse programs when you install the agent code.

The DSNUTILS program is a DB2[®] for z/OS[™] stored procedure that runs in a WLM and RRS environment. You can use the DSNUTILS program to run any installed DB2 utilities by using the user-defined stored procedure interface.

The definition of each warehouse program is added during initialization of the Data Warehouse Center. To see a list of the definitions, expand the **Programs and Transformers** folder in the main Data Warehouse Center window.

Transformer steps

Transformer steps are stored procedures and user-defined functions that specify statistical or warehouse transformers that you can use to transform data. You can use transformers to clean, invert, and pivot data; generate primary keys and period tables; and calculate various statistics.

In a transformer step, you specify one of the statistical or warehouse transformers. When you run the process, the transformer step writes data to one or more warehouse targets.

The two types of transformers are:

- Statistical transformers
- Warehouse transformers

In addition to the transformer steps, another transformer is a user-defined function. You can use the transformer with an SQL step.

The transformers for a particular operating system are packaged with the agent for that operating system. You install the warehouse programs when you install the agent code.

Restrictions:

The Data Warehouse Center transformers are only supported with a DB2 federated target database.

The definition of each transformer is added during initialization of the Data Warehouse Center. To see a list of the definitions, expand the **Programs and Transformers** folder in the main Data Warehouse Center window.

Before you use the transformers:

- Install the transformers in the target database.
- From the Database page of the Warehouse Target notebook, create the transformers and register them.
- From the Database page of the Warehouse Target notebook, enable the target database for the transformers.
- Change the program definition for each transformer to specify the agents and target resources.
- Ensure that the source and target tables are in the same database.

Replication steps

Replication steps copy designated changes in any DB2 relational databases from one location (a source) to another (a target), synchronizing the data in both locations. The source and target can be in logical servers (such as a

Defining and running processes

DB2 database or a DB2 for z/OS subsystem or data-sharing group) that are on the same computer or on different computers in a distributed network.

You can use replication steps to keep a warehouse table synchronized with an operational table without needing to fully load the table each time that the operational table is updated. With replication, you can use incremental updates to keep your data current.

Red Brick™ Warehouse utility program steps

Red Brick Warehouse utility program steps export data from and import data into a Red Brick Warehouse database.

User-defined program steps

A user-defined program step is a logical entity within the Data Warehouse Center that represents a business-specific transformation that you want the Data Warehouse Center to start. Because every business has unique data transformation requirements, businesses can choose to write their own program steps or to use tools like those provided by companies such as ETI or Vality.

For example, you can write a user-defined program that will perform the following functions:

1. Export data from a table.
2. Manipulate that data.
3. Write the data to a temporary output resource or a warehouse target.

If you require a function that is not supplied in one of these types of steps, you can write your own warehouse programs or transformers and define steps that use those programs or transformers.

Each group of steps has step subtypes. In all cases, you choose a specific step subtype to move or transform data. For example, the ANOVA transformer is a subtype of the Statistical transformer group.

Dependency checking for warehouse steps

Dependency checking ensures the integrity of the data in your warehouse by preventing steps from running when they might use damaged data. For example, you have two steps, Step 1 and Step 2, that are linked to a table. The table is a source table for Step 2 and a target table for Step 1. The table is not a transient table. If dependency checking is enabled for Step 2, and Step 1 fails or does not run, then you will receive an error message when Step 2 runs.

Dependency checking is available for steps that have sources that are populated by other steps in the warehouse. Dependency checking is enabled by default when you create a step; however, there are times when you might want to disable it. For example, you might want to disable dependency checking when you want a step to run regardless of the success or failure of the preceding step.

If you test a sequence of steps from the Process Model window with dependency checking enabled, you can set the retry property of the steps to 0. Setting the retry property to 0 causes errors related to dependency checking to be returned the Process Model window immediately. If the retry property is set to greater than 0, and dependency checking is enabled, errors related to dependency checking are returned to the Work in Progress window when all of the retry attempts are complete.

Related tasks:

- “Enabling warehouse transformers” in the *DB2 Warehouse Manager Standard Edition Installation Guide*
- “Running DB2 for z/OS utilities” in the *DB2 Warehouse Manager Standard Edition Installation Guide*

Warehouse file programs

The following table lists the warehouse file programs. Warehouse programs that manipulate files are in the warehouse file programs group. Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window.

Table 9. Agent sites and the warehouse file programs that they support

Name	Description	Windows	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
Copy file using FTP	Copies files on the agent site to and from a remote host.	✓	✓	✓	✓	✓
Run FTP command file	Runs any FTP command file that you specify.	✓	✓	✓	✓	✓
Data export with ODBC to file	Selects data in a table that is contained in a database registered in ODBC, and writes the data to a delimited file.	✓	✓	✓		
Submit z/OS JCL Jobstream	Submits a JCL jobstream to a z/OS system for processing.	✓	✓	✓		✓

DB2 warehouse utilities

The following table lists the DB2 warehouse utilities. Step subtypes are organized by utility group. A *utility group* is a logical grouping of related utilities. Utility groups are located on the left side of the Process Model window.

Table 10. DB2 warehouse utilities

Name	Description
DB2 UDB load	Loads data from a delimited file into a DB2 UDB database, either replacing or appending to existing data in the database.
DB2 data export	Exports data from a local DB2 database to a delimited file.
DB2 runstats	Runs the DB2 RUNSTATS utility on the specified table.

Defining and running processes

Table 10. DB2 warehouse utilities (continued)

Name	Description
DB2 reorg	Runs the DB2 REORG and RUNSTATS utilities on the specified table.

These DB2 warehouse utilities are supported on the following agent sites:

- AIX
- Linux
- Solaris Operating Environment
- Windows

The following DB2 warehouse utilities are specific to the iSeries agent site:

- DB2 for iSeries load replace
- DB2 for iSeries load insert

The following DB2 warehouse utilities are specific to the z/OS agent site:

- DB2 for z/OS load
- DB2 for z/OS runstats
- DB2 for z/OS reorg

The DB2 reorg warehouse utility is also available on the z/OS agent site.

Related concepts:

- “Warehouse processes” on page 87

Related tasks:

- “Defining the transformation and movement of data within the Data Warehouse Center” on page 88

DB2 OLAP Server programs

The following table lists the DB2 OLAP Server programs. Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window.

Table 11. DB2 OLAP Server programs

Name	Description
OLAP Server: Free text data load	Loads data from a comma-delimited flat file into a multidimensional DB2 OLAP Server database using free-form data loading.
OLAP Server: Load data from file with load rules	Loads data from a source flat file into a multidimensional DB2 OLAP Server database using load rules.
OLAP Server: Load data from SQL table with load rules	Loads data from an SQL table into a multidimensional DB2 OLAP Server database using load rules.
OLAP Server: Load data from a file without using load rules	Loads data from a flat file into a multidimensional OLAP Server database without using load rules.
OLAP Server: Update outline from file	Updates a DB2 OLAP Server outline from a source file using load rules.

Table 11. DB2 OLAP Server programs (continued)

Name	Description
OLAP Server: Update outline from SQL table	Updates a DB2 OLAP Server outline from an SQL table using load rules.
OLAP Server: Default calc	Calls the default DB2 OLAP Server calculation script associated with the target database.
OLAP Server: Calc with calc rules	Applies the specified calculation script to a DB2 OLAP Server database.

The following agent sites support these DB2 OLAP Server programs:

- AIX
- iSeries
- Linux
- Solaris Operating Environment
- Windows

These program groups are not supported on a z/OS agent site.

Related concepts:

- “Warehouse processes” on page 87

Related tasks:

- “Defining the transformation and movement of data within the Data Warehouse Center” on page 88

Replication programs

The following table lists the Replication programs. Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window.

Table 12. Replication programs

Name	Description
Base aggregate	Creates a target table that contains the aggregated data for a user table appended at specified intervals.
Change aggregate	Creates a target table that contains aggregated data based on changes recorded for a source table.
Point-in-time	Creates a target table that matches the source table, with a timestamp column added.
Staging table	Creates a consistent-change-data table that can be used as the source for updating data to multiple target tables.
User copy	Creates a target table that matches the source table at the time of the copy.

The following agent sites support these Replication programs:

- AIX
- iSeries
- Linux

Defining and running processes

- | • Solaris Operating Environment
- | • Windows
- | • z/OS

Related concepts:

- “Replication in the Data Warehouse Center” on page 135
- “Replication source tables in the Data Warehouse Center” on page 136
- “Replication password files” on page 140

Related tasks:

- “Setting up replication in the Data Warehouse Center” on page 137
- “Creating replication control tables” on page 138
- “Defining a replication step in the Data Warehouse Center” on page 139
- “Using a replication step in a process” on page 139

Warehouse deprecated programs

The following table lists the Visual Warehouse Version 5.2 warehouse programs and the DWC 7.2 Clean Data transformer. Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window. It is recommended that you do not use these programs or transformer to create new steps.

Because the command-line interface to some of the DB2 warehouse programs has changed from Visual Warehouse Version 5.2, the DB2 warehouse programs from Visual Warehouse Version 5.2 are supported separately.

Table 13. Agent sites and the Visual Warehouse Version 5.2 warehouse programs that they support

Name	Description	Windows	AIX or Linux	Solaris Operating Environment	iSeries
DB2 load replace (VWPLOADR)	Loads data from a delimited file into a DB2 database, replacing existing data in the database with new data.	✓	✓	✓	✓
DB2 load insert (VWPLOADI)	Loads data from a delimited file into a DB2 table, appending new data to the existing data in the database.	✓	✓	✓	✓
Load flat file into DB2 UDB Enterprise Server Edition (AIX only) (VWPLDPR)	Loads data from a delimited file into a DB2 Enterprise Server Edition database, replacing existing data in the database with new data.		✓		

Table 13. Agent sites and the Visual Warehouse Version 5.2 warehouse programs that they support (continued)

Name	Description	Windows	AIX or Linux	Solaris Operating Environment	iSeries
DB2 data export (VWPEXPT1)	Exports data from a local DB2 database to a delimited file.	✓	✓	✓	
DB2 runstats (VWPSTATS)	Runs the DB2 RUNSTATS utility on the specified table.	✓	✓	✓	
DB2 reorg (VWPREORG)	Runs the DB2 REORG and RUNSTATS utilities on the specified table.	✓	✓	✓	
DWC 7.2 Clean Data	Replaces data values, removes data rows, clips numeric values, performs numeric discretization, and removes white space.	✓	✓	✓	✓

The z/OS agent site supports only the DWC 7.2 Clean Data warehouse program.

Related concepts:

- “Data Warehouse Center Version 7.2 Clean Data transformer” on page 249
- “Rules tables for Data Warehouse Center Version 7.2 Clean Data transformer” on page 250

Related tasks:

- “Defining a Data Warehouse Center Version 7.2 Clean Data transformer” on page 252
- “Defining values for a Visual Warehouse 5.2 DB2 UDB Data Export (VWPEXPT1) program” on page 241
- “Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Insert (VWPLOADI) program” on page 242
- “Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Replace (VWPLOADR) program” on page 244
- “Defining values for a Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program” on page 246
- “Defining values for a Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) program” on page 247
- “Defining values for a Visual Warehouse 5.2 Load flat file into DB2 Enterprise Server Edition (VWPLDPR) program (AIX)” on page 247

Warehouse transformers

The following table lists the warehouse transformers. Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window.

Table 14. Warehouse transformers

Name	Description
Clean data	Replaces data values, removes data rows, clips numeric values, performs numeric discretization, carries over columns, encodes invalid values, changes case, performs error processing.
Generate key table	Generates or modifies a sequence of unique key values in an existing table.
Generate period table	Creates a table with generated date, time, or timestamp values, and optional columns based on specified parameters, or on the date or time value, or both, for the row.
Invert data	Inverts the rows and columns of a table, making rows become columns and columns become rows.
Pivot data	Groups related data from selected columns in a source table into a single column in a target table. The data from the source table is assigned a particular data group in the output table.

These warehouse transformers are supported on the following agent sites:

- AIX
- iSeries
- Linux
- Solaris Operating Environment
- Windows
- z/OS

However, the Generate key table transformer is not supported on a z/OS agent site.

Related tasks:

- “Transforming target tables” on page 143
- “Steps and tasks: Data Warehouse Center help”

Statistical transformers

The following table lists statistical transformers. Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window.

Table 15. Statistical transformers

Name	Description
ANOVA	Computes one-way, two-way, and three-way analysis of variance; estimates variability between and within groups and calculates the ratio of the estimates; and calculates the p-value.

Table 15. Statistical transformers (continued)

Name	Description
Calculate statistics	Calculates count, sum, average, variance, standard deviation, standard error, minimum, maximum, range, and coefficient of variation on data columns from a single table.
Calculate subtotals	Uses a table with a primary key to calculate the running subtotal for numeric values grouped by a period of time, either weekly, semimonthly, monthly, quarterly, or annually.
Chi-square	Performs the chi-square and chi-square goodness-of-fit tests to determine the relationship between values of two variables, and whether the distribution of values meets expectations.
Correlation	Computes the association between changes in two attributes by calculating correlation coefficient r, covariance, T-value, and P-value on any number of input column pairs.
Moving average	Calculates a simple moving average, an exponential moving average, or a rolling sum, redistributing events to remove noise, random occurrences, and large peaks or valleys from the data.
Regression	Shows the relationships between two different variables and shows how closely the variables are correlated by performing a backward, full-model regression.

These programs are supported on the following agent sites:

- AIX
- iSeries
- Linux
- Solaris Operating Environment
- Windows
- z/OS

Related concepts:

- “ANOVA transformer” on page 163
- “Calculate Statistics transformer” on page 164
- “Calculate Subtotals transformer” on page 164
- “Chi-square transformer” on page 166
- “Moving averages” on page 168
- “Regression transformer” on page 169

Related tasks:

- “Steps and tasks: Data Warehouse Center help”

User-defined programs

A user-defined program is the Format date and time program. This program changes the format of the date and time in a source table date field.

This program is supported on the following agent sites:

- AIX
- Linux
- Solaris Operating Environment

Defining and running processes

- Windows

Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window.

Related concepts:

- “Example: Specifying pre-defined parameters for a user-defined program” on page 193
- “Parameters for user-defined programs” on page 191
- “Agent sites for user-defined programs” on page 190
- “Trillium Batch System user-defined program” on page 158
- “MQSeries stored procedure” on page 178
- “What is a user-defined program?” on page 189
- “Status information for user-defined programs” on page 194
- “Example: Passing parameters to a user-defined program” on page 192

Related tasks:

- “Enabling a user-defined program (Windows)” on page 192
- “Using a user-defined program in the Data Warehouse Center” on page 190

Related reference:

- “Error return codes for the MQXMLXF stored procedure” on page 178

Red Brick Warehouse utility programs

The following table lists the Red Brick warehouse utility programs. These utility programs export data from and import data to a Red Brick Warehouse database. Step subtypes are organized by program group. A *program group* is a logical grouping of related programs. Program groups are located on the left side of the Process Model window. The following table shows agent sites and the Red Brick warehouse utility programs that they support.

Table 16. Supported Red Brick warehouse utility programs

Name	Description	Windows	AIX or Linux	Solaris Operating Environment
TMU Load	Loads data from an external file into a Red Brick database table.	✓	✓	✓
SQL Export	Exports data from Red Brick database tables or views to a file in delimited format or fixed column format.	✓	✓	✓

Column mapping

The Data Warehouse Center makes manipulating data easy. You decide which rows and columns (or fields) in the source database that you will use in your warehouse database. Then, you define those rows and columns in your step.

For example, you want to create some steps that are related to manufacturing data. Each manufacturing site maintains a relational database that describes the products that are manufactured at that site. You create one step for each of the four sites.

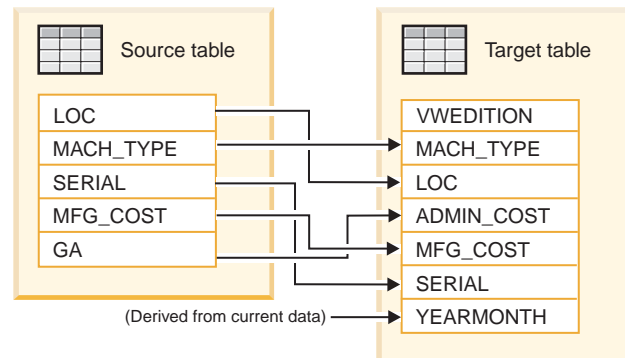


Figure 8. An example of mapping source data to a warehouse table

Only certain steps use column mapping. If the column mapping page is blank after you define parameter values for your step that result in more than one column, then your step does not use column mapping.

On the Column Mapping page of the step notebook, map the output columns that result from the transformations that you defined on the Parameters page, or the SQL statement page, to columns on your target table. On this page, output columns from the Parameters page are referred to as source columns. Source columns are listed on the left side of the page. Target columns from the output table linked to the step are listed on the right side of the page. On the Column Mapping page, you can perform the following tasks:

- Create a mapping by clicking a source column and dragging it to a target column. An arrow is drawn between the source column and the target column.
- Map all of the unmapped columns by clicking **Actions** → **Map All** and selecting one of the following options:
 - By Name Only**
Source columns are mapped to target columns with the same name.
 - By Name and Type**
Source columns are mapped to target columns with matching names and compatible data types.
 - By Position**
Source columns are mapped to target columns based on their position in the table.
- Remove a mapping by right-clicking an arrow, and clicking **Remove**.
- Remove all mappings by clicking **Actions** → **Unmap All**.
- Add columns to the target column list by clicking **Actions** → **Add Columns**. This option is only available if there are unmapped source columns and no unmapped matching columns in the target column list.
- Locate columns in the source columns list, target columns list, or both by clicking **Actions** → **Find Columns**. Wildcard characters are not supported.

A target column that is created during the current session of the Properties notebook for the step can be edited or removed on the Column Mapping page. After you click **OK** to exit the step, you cannot edit those columns on the Column Mapping page during subsequent sessions. Instead, you must edit the target

Defining and running processes

columns in the target table object. To change the attributes of a target column on the Column Mapping page, click the attribute and type or select the new value.

Some step subtypes allow you to create a default target table based on the parameters you supply for the step.

For certain step subtypes, the actions that you can perform on this page are limited. For other step subtypes, the column outputs from the Parameters page follow certain rules.

Related tasks:

- “Defining statistical transformers in the Data Warehouse Center” on page 163
- “Creating a column mapping: Data Warehouse Center help”
- “Removing a column mapping: Data Warehouse Center help”

Related reference:

- “Warehouse transformers” on page 96
- “Statistical transformers” on page 96

Running warehouse steps

This section covers:

- Step promotion and testing
- Setting the processing options for a step
- Verifying the results of a step
- Running steps from outside the Data Warehouse Center
- Printing step information to a text file

Step promotion and testing

The Data Warehouse Center lets you manage the development of your steps by classifying steps in one of three modes: development, test, or production. The mode determines whether you can make changes to the step and whether the Data Warehouse Center will run the step according to its schedule. *Promoting* a step means moving the step to a higher mode. You cannot promote shortcuts to steps that were created in other processes. The Promote menu choice is unavailable when you select multiple objects to promote if a shortcut is selected.

Development mode:

When you first create a step, it is in development mode. You can change any of the step properties in this mode. A table for the step does not exist in the target warehouse. You cannot run the step to test it, and the Data Warehouse Center will not run the step according to its automated schedule.

Test mode:

You run steps to populate their targets with data. You can then verify whether the results are what you expected.

Before you run steps, you must promote them to test mode.

In the step properties, you can specify that the Data Warehouse Center is to create a target table for the step. When you promote the step to test mode, the Data Warehouse Center creates the target table. Therefore, after you promote a step to test mode, you can make only those changes that are not destructive to the target table. For example, you can add columns to a target table when its associated step is in test mode, but you cannot remove columns from the target table.

After you promote steps to test mode, you run each step individually. The Data Warehouse Center will not run the step according to its automated schedule.

Production mode:

To activate the schedule and the task flow links that you created, you must promote steps to production mode. Production mode indicates that steps are in their final format. In production mode, you can change only the settings that will not affect the data that is produced by the step. You can change schedules, processing options (except population type), or descriptive data about the step. You cannot change the parameters of the step.

Related tasks:

- “Promoting a step: Data Warehouse Center help”
- “Testing a step: Data Warehouse Center help”

Verifying the results of a step that is run in test mode

After you promote a step to test mode, you can use the DB2 Control Center to verify that the target table was created.

Procedure:

To verify that the target table was created:

1. Click **Tools** → **Control Center** to open the DB2 Control Center from the main Data Warehouse Center window.
2. In the DB2 Control Center, expand the object tree until you see the warehouse database.
3. Expand the database.
4. Expand the **Tables** folder to see a list of tables displayed in the Contents pane of the window.
5. Verify that the target table is in the list.

After you run the step, you can use the Sample Contents function to view the data that was moved.

Running steps that use transient tables as sources

You can run steps in a process that use a transient table as a data source. Transient tables contain temporary data. After the data is used by the steps, it is purged from the table. You

Prerequisites:

You must have more than one step to run a process with a transient table.

Procedure:

Defining and running processes

To run steps that use transient tables as sources:

1. Set up your steps so that the first step that runs populates the transient table and the second step that runs uses the transient table as a source.
2. Promote both steps to production mode.
3. Schedule the steps to run at a specific date and time, or run the steps manually.

When the process runs, it will skip the first step because the step is linked to the transient table. When the second step begins, there is no data in the transient table, so the process will run the first step to populate the table, and then run the second step. When the second step is complete, all of the rows in the transient table are deleted.

You can change a transient table to a permanent table only when the step is in development mode. Transient tables cannot be changed to permanent tables in production mode.

Running a step from outside the Data Warehouse Center using the external trigger program

You can start a step independently of the Data Warehouse Center administrative interface by using an external trigger program. An *external trigger program* is a warehouse program that calls the Data Warehouse Center.

The external trigger program consists of two components: the external trigger server (XTServer) and the external trigger client (XTClient). The XTServer is installed with the warehouse server. The XTClient is installed with the warehouse agent for all agent types.

Prerequisites:

- A step must have the **Run on demand** option specified on the Processing Options page of the Properties notebook for the step before it can be triggered by the external trigger program.
- If you are using an external trigger program from a version of DB2 Universal Database that is older than DB2 Enterprise Server Edition Version 8, you must update your CLASSPATH variable to point to the correct Java class files. See the following table for the correct Java class files for your operating system:

Operating system	CLASSPATH variable
iSeries	/qibm/userdata/os400/sqllib/Function/JET
Unix	sqllib/tools/db2XTrigger.jar, sqllib/tools/db2vwcom.jar, and sqllib/java/common.jar
Windows	sqllib\tools\db2XTrigger.jar, sqllib\tools\db2vwcom.jar, and sqllib\java\common.jar
z/OS	Point to the directory where the Data Warehouse Center is installed. For example, /user/lpp/dwc81

Figure 9. CLASSPATH variables for different operating systems

- To use the external trigger program to start a step, you must have the Java™ 2 Platform, Standard Edition Version 1.3 installed on the warehouse server workstation and the agent site. You can also use the IBM Software Developer's Kit that is installed with the Data Warehouse Center and the Control Center.

Procedure:

To start the external trigger program:

1. Start the external trigger server.
2. Start the external trigger client.

Related reference:

- “Syntax for starting the external trigger server” on page 103
- “Syntax for starting the external trigger client” on page 103

Syntax for starting the external trigger server

You must start the external trigger server before you issue commands to the external trigger client. This must be done on the warehouse server.

XTServer

```
▶▶—java—db2_vw_xt.XTServer—TriggerServerPort—————▶▶
```

TriggerServerPort

The TCP/IP port assigned to the external trigger server.

This value can be 11004 or any available port number.

Syntax for starting the external trigger client

```
▶▶—java—[—DDWC_MIN=min_port_number] [—DDWC_MAX=max_port_number]—————▶▶
▶▶—db2_vw_xt.XTClient—TriggerServerHostName—TriggerServerPort—DWCUserID—————▶▶
▶▶—DWCUserPassword—StepOrProcessName—Command—[—WaitForStepCompletion] [—RowLimit] [—LogToFile]—————▶▶
```

-DDWC_MIN=min_port_number

Optional. The minimum port number that the firewall allows for the XTClient to communicate with the XTServer. You must specify a value for this parameter to enable firewall support for the XTClient.

-DDWC_MAX=max_port_number

Optional. The maximum port number that the firewall allows for the XTClient to communicate with the XTServer. You must specify a value for this parameter to enable firewall support for the XTClient.

TriggerServerHostName

The TCP/IP host name for the workstation on which the XTServer is running.

Specify a fully qualified host name.

TriggerServerPort

The TCP/IP port assigned to the XTServer server. The external trigger client must use the same port as the external trigger server.

This value must be the one that the XTServer is running on.

DWCUserID

A user ID with Data Warehouse Center operations privileges.

Defining and running processes

DWCUserPassword

The password for the user ID.

StepOrProcessName

The name of the step or process to start.

The name is case-sensitive. Enclose the name in quotation marks if it includes blanks, such as "Corporate Profit".

Command

One of the following values:

1 Populate (or run a step).

The user ID under which you run the external trigger program must be in the same warehouse group as the process that contains the step.

2 Promote a step to test mode.

The user ID under which you run the external trigger program must be in the same warehouse group as the process that contains the step.

3 Promote a step to production mode.

The user ID under which you run the external trigger program must be in the same warehouse group as the process that contains the step.

4 Demote a step to test mode.

The user ID under which you run the external trigger program must be in the same warehouse group as the process that contains the step.

5 Demote a step to development mode.

The user ID under which you run the external trigger program must be in the same warehouse group as the process that contains the step.

6 Populate a process.

The process must be enabled to be populated.

7 Check to See Whether DWC Server is Running.

Verify that the Data Warehouse Center server is running.

WaitForStepCompletion

Optional. This parameter indicates whether the external trigger program is to return the results of the step or process processing. Choose one of the following values:

1 Wait for the step or process to complete. Return 0 if the step or process completes successfully, or return an error if the step or process fails.

0 or blank

Do not wait for the step or process to complete.

RowLimit

Optional. This parameter indicates whether the external trigger program is to extract all the rows of the source table or a subset of the rows. You can

use this parameter to quickly test the steps that extract data from large databases. Choose one of the following values:

0 Fetch all rows

n Fetch *n* rows

This parameter is valid only when the step is in test mode.

LogToFile

Optional. This parameter indicates whether the external trigger program is to log traces to the XTClient.log file. Choose one of the following values:

1 or 0 Log traces to the XTClient.log file

The CLASSPATH variable must be set correctly before you run the command.

Example

For example, you want to start the Corporate Profit step using a user ID of db2admin and a password of db2admin. The external trigger program is on the dwserver host. You issue the following command:

```
java XTClient dwserver 11004 db2admin db2admin "Corporate Profit" 1
```

Related tasks:

- “Running a step from outside the Data Warehouse Center using the external trigger program” on page 102

Related reference:

- “Syntax for starting the external trigger server” on page 103

Data Warehouse Center messages for the external trigger program

When you run the external trigger program, it sends a message to the warehouse server. If the message is sent successfully, the external trigger program returns a zero return code.

If you specified a value of 1 for the *WaitForStepCompletion* parameter, the external trigger program waits until the step or process finishes running and then returns the return code from that run.

The external trigger program returns a nonzero return code if it cannot send the message to the warehouse server. The return codes match the corresponding codes that are issued by the Data Warehouse Center function when there is a communications error or when authentication fails.

Related tasks:

- “Running a step from outside the Data Warehouse Center using the external trigger program” on page 102

Related reference:

- “Syntax for starting the external trigger server” on page 103
- “Syntax for starting the external trigger client” on page 103

Printing step information to a text file

You can print information about a step (such as subject area, source table names, and target table names) to a text file.

Procedure:

To print information about a step to a text file:

1. Right-click the step icon in the Process Model window.
2. Click **Print**—> **Print to File**.
3. Specify the name of the file where you want to print the information.

Related tasks:

- “Steps and tasks: Data Warehouse Center help”

Promoting warehouse processes

You can promote or demote all of the steps in a process to the same mode by promoting or demoting the process. Only the schedules and cascades for the steps are affected. When the steps are in production their cascades and schedules will be active. In order for the process schedules and cascades to be active the process must be enabled.

Managing step task flow

To schedule data movement and transformation, you can use the Data Warehouse Center scheduling capabilities, or you can use the Data Warehouse Center with another product that provides scheduling functions.

Step task flow

In the Data Warehouse Center, you can start a step in two ways. You can specify that a step is to start after another step has run, or you can schedule the step to start at a specified date and time. You can combine these methods to run the steps in a process. You can schedule the first step to run at a specified date and time. These schedules and cascades are active when the step is in production. After you schedule the first step, you can specify that another step is to start after the first step runs, and specify that a third step is to start after the second step runs, and so on.

If you use more than one condition to define the task flow between two steps, the condition becomes On completion when you close and then reopen the Process Model window. For example, you might define the following task flow for Step A and Step B on the Step Task Flow page of the Schedule notebook:

```
Step A -> On success -> Step B  
Step A -> On failure -> Step B
```

Then save the step, and close and reopen the Process Model window. The Step Task Flow page in the Schedule notebook shows the following task flow for Process A and Process B:

```
Step A -> On completion -> Step B
```

Related concepts:

- “Process task flow” on page 107

Related tasks:

- “Creating cascade relationships: Data Warehouse Center help”
- “Creating cascade relationships for steps: Data Warehouse Center help”
- “Creating cascade relationships for processes: Data Warehouse Center help”
- “Steps and tasks: Data Warehouse Center help”

Cascading processes

This topic describes how to schedule and set up task flow for warehouse processes.

Process task flow

In the Data Warehouse Center, you can schedule processes as well as steps. You can specify that a process is to start after another process has run. You must carefully group your steps into a meaningful process, so that you can schedule and specify the task flow of your processes properly. You can start a process based on the completion of another process using the Process Task Flow page of the Scheduler notebook. A process is determined to be complete based on the following conditions:

On success

If the process has a cascade-on-success condition, then all terminal steps within the process must complete with success or warning return codes in order for the cascade to run. Steps outside the process that are triggered by steps inside the process are not a factor in determining the success or failure of a process.

On failure

The process will run if the terminal step in the preceding process fails. Steps outside the process that are triggered by steps inside the process are not a factor in determining the success or failure of a process.

On completion

When the steps in the preceding process have run, the next process will run regardless of the success or failure of the steps in the preceding process.

If you use more than one condition to define the task flow between two processes, the condition becomes On completion when you close and then reopen the Process Model window. For example, you might define the following task flow for Process A and Process B is defined on the Process Task Flow page of the Schedule notebook:

```
Process A -> On success -> Process B
Process A -> On failure -> Process B
```

Then save the process, and close and reopen the Process Model window. The Process Task Flow page shows the following task flow for Process A and Process B:

```
Process A -> On completion -> Process B
```

Only the steps that are in production mode are run in a cascading process.

Processes scheduling

In addition to being able to define the task flow of a process, you can also schedule a process to run on a particular date and time, or on a particular schedule of dates and times. Use the Schedule page in the Scheduler notebook to schedule a process.

Defining and running processes

Steps in the process can have schedules that are different from the process schedule. A step will not run if it is scheduled to run while it is already running as part of the process. If the step is scheduled to run before or after it has run as part of the process, then it will run even if the process has not completed. You can make changes to the steps in the process regardless of whether the process schedule is enabled or disabled.

When you group your steps in a process, do not to use steps that refer to themselves. Self-referencing steps cause the process to enter an infinite loop.

Shortcuts in cascading processes are run, but they do not affect the completion status of a process.

Related concepts:

- “Warehouse processes” on page 87

Related tasks:

- “Creating cascade relationships for processes: Data Warehouse Center help”
- “Purging steps and processes from the Work in Progress window: Data Warehouse Center help”

Monitoring the progress of steps and processes

You can monitor the progress of steps and process by using the Work in Progress window, trace files, and sampling the data in your target tables after a step or process has run.

Monitoring the progress of a step or process using the Work in Progress window

You can use the Work in Progress window to monitor the progress of all steps and processes in the Data Warehouse Center that are running or scheduled to run. The Work in Progress window shows an entry for the step or process that is running. While the step is running, it has a status of Populating. If the processing fails, you can use the **Show Log** action to find the problem.

In the Work In Progress window, success for steps represents an edition of data. The edition can be purged or rerun using the Run Now action. Purging or rerunning the edition changes the data in the edition. Each step and process that is run has an entry in the Work In Progress window. Only one entry per process and step edition are kept. All records have log entries.

Procedure:

To monitor the progress of steps and processes:

1. Open the Work In Progress window from the main Data Warehouse Center window.
2. Click **Data Warehouse Center** —> **Work in Progress**.

Related tasks:

- “Viewing log entries in the Data Warehouse Center” on page 228
- “Work in Progress -- Overview: Data Warehouse Center help”
- “Canceling steps that are currently running: Data Warehouse Center help”

- “Purging steps and processes from the Work in Progress window: Data Warehouse Center help”
- “Rerunning steps from the Work in Progress window: Data Warehouse Center help”
- “Running steps from the Work in Progress window: Data Warehouse Center help”
- “Viewing the current status of a step or process: Data Warehouse Center help”

Step and process error messages

Errors that occur when a step runs are recorded in the log entries in the Work in Progress window and can be viewed using the Log Viewer. Errors that occur when a step is promoted can be viewed by using the Show Log action in the Work in Progress window.

For user-defined programs, if the **Error RC1** field has a value of 8410, then the program failed during processing. Find the value of the **Error RC2** field, which is the value returned from the program, by looking through the log entries.

Look in the trace files for more information about the program processing.

Related concepts:

- “Warehouse log files” on page 227
- “Error logging for warehouse programs and transformers” on page 231

Related tasks:

- “Viewing build-time errors using the basic logging function” on page 228
- “Viewing log entries in the Data Warehouse Center” on page 228
- “Viewing run-time errors using the basic logging function” on page 227

Sampling data

Use the Sample Contents function to verify the data in a table in the Data Warehouse Center. The Sample Contents function uses the first agent site returned from the metadata.

Restrictions:

Flat files cannot be accessed by Sample Contents.

Procedure:

To sample data:

1. From the Process Model window or the DB2 Control Center, right-click the target table.
2. Click **Sample Contents** to see a subset of the data in the table.

Defining and running processes

Chapter 9. Selecting, inserting, and updating source data in a target table

The Data Warehouse Center provides several different ways to move data, depending on your needs. This chapter explains how to use SQL steps to:

- Select source data and insert it into a target table
- Select source data and update it in the target table
- Join columns in tables

SQL steps

You can use an SQL Select and Insert step to select source columns and insert the data from the columns into a target table. You can use an SQL Select and Update step to update data in a target table. You can specify that the Data Warehouse Center create the target table based on the source data or use the source data to update an existing table.

You can use a warehouse source or a warehouse target as a source for an SQL step.

Tip: When you create editioned SQL steps based on usage, you might want to consider creating a non-unique index on the edition column to speed performance of deleting of editions. Consider this for large warehouse tables only, because the performance of inserts can be impacted when inserting a small number of rows.

Related tasks:

- “Defining an SQL step” on page 111
- “Steps and tasks: Data Warehouse Center help”

Defining an SQL step

Use the Select and Insert SQL step to select and insert data into a target table. Use the Select and Update SQL step to update the data in a target table.

Prerequisites:

You must link the step to a source and a target in the Process Model window before you define its properties.

Restrictions:

You cannot change an SQL step that is in production mode.

Procedure:

To define an SQL step, add the SQL step that you want to use to a process, open the Properties notebook for the step, and define the step properties.

Related tasks:

- “Steps and tasks: Data Warehouse Center help”

Incremental commit

Incremental commit is an option that is available for Select and Insert SQL steps that allows you to control the commit scope of the data that is managed by the Data Warehouse Center. Incremental commit can be used when the volume of data to be moved by the agent is large enough that the DB2[®] log files might fill up before the entire work of the step is complete, or when you want to save partial data. SQL steps will complete with an error if the amount of data being moved exceeds the DB2 maximum log files that have been allocated. The performance of your database can deteriorate because a significant number of commits can occur when you use incremental commit.

You use the incremental commit option to specify the number of rows (rounded to the nearest factor of 16) to be processed before a commit is performed. The agent selects and inserts data, committing incrementally, until it completes the data movement successfully. When the data movement completes successfully, outdated editions are removed (if the target has editions). Tables that use incremental commit and have editions might require additional space for the table, because the outdated edition is not deleted until the new edition is successfully inserted.

If the source database and target database are the same, the SQL cannot be optimized. If you specify incremental commit, the Data Warehouse Center uses Select and Insert logic rather than using the optimized Insert from Select.

In a step where the population type is Replace, use the incremental commit only when the new data is inserted. The old data is deleted within a single commit scope. If you need to delete the data without producing log records, run a step that loads an empty file before you run the SQL step with the population type of Append. Steps with a population type of Append with no edition can contain partial data if an error occurs after a commit is performed. Steps with a population type of Append with editions can contain partial data if an error occurs after a commit is performed, but the outdated editions will not be deleted.

If an error occurs after an incremental commit is performed, evaluate the failure and take the appropriate action to start the SQL step again at a logical point. The committed data will remain in the target table until you run the step again. If an error occurs when a step is populating a target database and the value of incremental commit is greater than 0, all of the results that were committed before the error occurred will appear in the target database.

Joining source columns

This section describes how to join the columns in source tables.

Defining a join

If you select multiple sources, you need to specify how the rows of these columns will be associated in the join of the tables. You can use SQL Assist to define the join.

Prerequisites:

You must link the sources to the step before you define the join.

Procedure:

Selecting, inserting, and updating source data in a target table

To join source tables:

1. Define an SQL step.
2. Open the step.
3. Click the **SQL Statement** tab.
4. Click **Build SQL** to automatically generate SQL. (Or you can type your own SQL.)
SQL Assist opens.
5. Click the **Tables** tab and verify that the sources are listed.
6. Click the **Columns** tab.
7. From the **Available columns** list, select a column, and click > to move the column to the **Selected columns** list. Click >> to select and move all of the columns.
8. Click the **Joins** tab.
9. Select a column in one of the tables. The tables are displayed in the order that they are shown in the **Selected tables** list on the Tables page.
10. Select a column in another table.
If the columns have compatible data types, a gray line is displayed, connecting the columns, and the **Join** push button is available.
If the columns do not have compatible data types, an error message is displayed in the status area at the bottom of the window.
11. Click the **Join Type** push button to create the join.
SQL Assist draws a red line between the selected columns, which indicates that the tables are joined on that column.
12. To create additional joins, repeat the previous steps.
13. Click the **Review** tab to view the SQL statement that you just built.
14. Click **OK** to save your changes and close SQL Assist.
15. Click **OK** to save your changes and close the step Properties notebook.

Related concepts:

- “Join types” on page 114
- “Star joins” on page 116

Related tasks:

- “Removing a join” on page 113

Removing a join

You can remove a join by using the Build SQL notebook in the Data Warehouse Center.

Procedure:

To remove a join:

1. Open the properties notebook for the SQL step.
2. Click the **SQL Statement** tab.
3. Click **Build SQL**.
4. Click the **Joins** tab.
5. Select the joined columns. A red line indicates the currently selected join. Other joins are indicated by blue lines.

Selecting, inserting, and updating source data in a target table

6. Click **Unjoin**. The join line is removed.

Related concepts:

- “Star joins” on page 116

Related tasks:

- “Defining a join” on page 112

Transforming codes

In many production environments, the source data includes encoded information. For example, you might use codes to refer to part numbers throughout your database. You also have a table that relates the part numbers to serial numbers and descriptions of each part. In these situations, you want the information in your warehouse to include the name and description of each part. To do this, you must combine the decoding table with the source data that contains the encoded part numbers.

Procedure:

To combine the decoding table with the source data that contains the encoded part numbers:

1. Define the decoding table and the encoded part numbers table as part of a warehouse source.
2. Select these tables as source tables for a step.
3. Click **Join** on the Joins page of SQL Assist to join the tables.

Alternatively, you can use a CASE statement to decode the data. For example, data in a month column is coded numerically, and you want to transform the data to character strings that contain the abbreviation of the month. You issue the following statement:

```
CASE TBC.ORDER_HISTORY.ORDERMONTH WHEN 1 THEN 'Jan'  
WHEN 2 THEN 'Feb' WHEN 3 THEN 'Mar' WHEN 4 THEN 'Apr'  
WHEN 5 THEN 'May' WHEN 6 THEN 'Jun' WHEN 7 THEN 'Jul'  
WHEN 8 THEN 'Aug' WHEN 9 THEN 'Sep' WHEN 10 THEN 'Oct'  
WHEN 11 THEN 'Nov' WHEN 12 THEN 'Dec'  
END
```

Related concepts:

- “Join types” on page 114
- “Star joins” on page 116

Join types

By default, a join is assumed to be an inner join. You can also request other types of joins by clicking **Join Type** on the Joins page of SQL Assist. The following types of joins are available:

- Inner join
- Left outer join
- Right outer join
- Full outer join

Selecting, inserting, and updating source data in a target table

An *inner join* is a join method in which a column that is not common to all of the tables being joined is dropped from the resultant table. If your database supports the OUTER JOIN keywords, you can extend the inner join to add rows from one table that have no matching rows in the other table.

For example, you want to join two tables to get the last name of the manager for each department. The first table is a Department table that lists the employee number of each department manager. The second table is an Employee table that lists the employee number and last name of each employee. However, some departments do not have a manager; in these cases, the employee number of the department manager is null. To include all departments regardless of whether they have a manager, and the last name of the manager, if one exists, you create a *left outer join*. The left outer join includes rows in the first table that match the second table or are null. The resulting SQL statement is as follows:

```
SELECT DEPTNO, DEPTNAME, EMPNO, LASTNAME
FROM DEPARTMENT LEFT OUTER JOIN EMPLOYEE
ON MGRNO = EMPNO
```

A *right outer join* is the same as a left outer join, except that it includes rows in the second table that match the first table or are null. A *full outer join* includes matching rows and null rows from both tables.

For example, you have two tables, Table 1 and Table 2, with the following data:

Table 17. Table 1

Column A	Column B
1	A
2	B
3	C

Table 18. Table 2

Column C	Column D
2	X
4	2

You specify a join condition of Column A = Column C. The result tables for the different types of joins are as follows:

Inner join

Table 19. Inner join result table

Column A	Column B	Column C	Column D
2	B	2	X

Left outer join

Table 20. Left outer join result table

Column A	Column B	Column C	Column D
1	A	null	null
2	B	2	X
3	C	null	null

Selecting, inserting, and updating source data in a target table

Right outer join

Table 21. Right outer join result table

Column A	Column B	Column C	Column D
2	B	2	X
null	null	4	2

Full outer join

Table 22. Full outer join result table

Column A	Column B	Column C	Column D
1	A	null	null
2	B	2	X
3	C	null	null
null	null	4	2

If you specify value (a,c), you obtain the following result:

Table 23. Result of value (a,c)

Value (a,c)
1
2
3
4

Related concepts:

- “Star joins” on page 116

Related tasks:

- “Defining a join” on page 112
- “Removing a join” on page 113
- “Transforming codes” on page 114

Star joins

You can generate a *star join*, which is a join of source tables that are defined in a star schema. A *star schema* is a specialized design that consists of the following types of tables:

- *Dimension tables*, which describe aspects of a business
- A *fact table*, which contains the facts about the business

For example, if you have a mail-order business that sells books, some dimension tables are Customers, Books, Catalogs, and Fiscal_Years. The fact table contains information about the books that were ordered from each catalog by each customer during the fiscal year.

Selecting, inserting, and updating source data in a target table

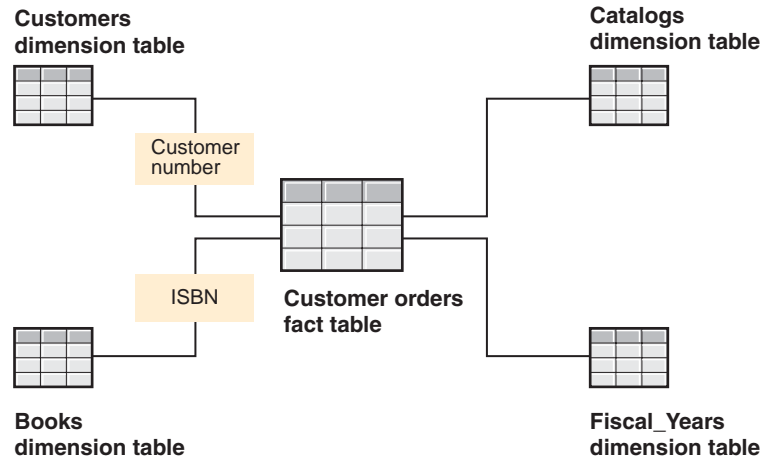


Figure 10. A star schema

Each dimension table contains a *primary key*, which is one or more columns that you select to identify a row in the table. The fact table contains *foreign keys* that correspond to the primary keys in the dimension table. A foreign key is a column in a table whose allowable values must exist as the primary key for another table.

When you request a star join, the Data Warehouse Center joins the primary keys of the dimension tables with the foreign keys of the fact table. In the previous example, the Customers table has a primary key of Customer Number, and each book has a primary key of its Book Number (ISBN). Each order in each table contains foreign keys of Customer Number and Book Number. The star join combines information about the customers and books with the orders.

Filtering data

In most cases, when you create a step, you want only a subset of the source data. You might want to extract only the rows that meet certain criteria. You can use the Data Warehouse Center to build an SQL WHERE clause to limit the rows that you extract from the source table.

For example, you can define a step that selects rows from the most recent edition of the source table:

```
WHERE TBC.ORDER_HISTORY.RUN_ID = &cur_edtn.IWHDATA.TBC.ORDER_HISTORY
```

The RUN_ID column contains information about the step edition. The &cur_edtn token represents the current step edition. Therefore, this WHERE clause selects rows in which the step edition equals the current edition.

Procedure:

To build the WHERE clause, use the Conditions page of SQL Assist.

To exclude duplicate rows, select the **Exclude duplicate rows (SELECT DISTINCT)** check box. This action eliminates all but one of each set of duplicate rows in the result set.

To build complex conditions, click **Advanced Expression**. The Expression Builder - Conditions window opens.

Selecting, inserting, and updating source data in a target table

| You can remove search conditions by highlighting the portion of the condition that
| you want to remove in the **Conditions** field, and pressing the Delete key.

Specifying search conditions when filtering data

When you are defining an SQL step and specifying data, you can specify search conditions.

Procedure:

To specify search conditions:

1. From the SQL Statement page of the Step Properties notebook, click **Build SQL**. SQL Assist opens.
2. Click the **Conditions** tab.
3. Select the column for the search from the **Available columns** list.
4. Select an operator from the **Operators** list.
5. Type one or more values in the **Values** list. Enter one value per line. Click **Clear** to remove all of the values from the **Values** list.

If you selected the Between operator in the **Operators** list, you must enter two values per line in the **Values** list. The values are separated by an ampersand (&).

You can search for appropriate values by clicking **Find**.

You can specify host variables in the **Values** list:

- a. Click on a line in the **Values** list.
- b. Click **Add Variable** to open the Add Variable window.
- c. Type the variable name.
- d. Click **OK** to close the Add Variable window and add the variable to the **Values** list on the **Conditions** page.

You can specify parameters in the **Values** list. If you specify a parameter, its value is used in the search condition. You specify a parameter in the format :parm, where parm is the parameter name. For example, :empid is a valid specification for a parameter named empid.

6. Click **Add** to add the condition to the **Conditions** field.
7. To specify additional search conditions for your statement, click **And** or **Or**. Then, build the second search condition as described in the previous steps. When you click **Add**, the subsequent conditions are appended to those that already are displayed in the **Conditions** field.

Creating composite summary steps

The Data Warehouse Center lets you easily and accurately define steps that are summaries of source data. You can use standard SQL aggregation functions (AVG, COUNT, MAX, MIN, and SUM) and the SQL GROUP BY clause to create steps that summarize the source data.

Summary steps reduce the load on the network. They perform the aggregations on the source data before replicating the data across the network. You can also create composite steps that use summary techniques to summarize other steps. Summarizing reduces the size of the target warehouse that you create.

Procedure:

Selecting, inserting, and updating source data in a target table

To create a step with a composite summary, click the **SUM** function in the **Functions** field of the Expression Builder window of SQL Assist.

For example, a step summarizes all the items that are sold in a month and expresses the amount in thousands of dollars:

```
SUM(TBC.ITEMS_MONTH.Amount)/1000
```

Related tasks:

- “Defining an SQL step” on page 111

Defining a calculated column and adding it to a step

You can define some columns that are calculated from values of other columns. For example, you need only the month in which an item was ordered. You can use the SQL **DATE** function to convert the order date to the **DATE** data type format. Then, you use the **MONTH** function to return the month part of the date. The SQL statement for the calculated column is as follows:

```
MONTH(DATE(TBC.ORDERS_MONTH.OrderDate))
```

You can also use calculated columns to summarize data. In many situations, your source data contains more detail than you want to replicate into your warehouse. All you need from the source data is a summary of some kind. You might need an average, a summary, or a count of the elements in the source database, but you do not need all the data.

Procedure:

To define a calculated column:

1. From the SQL Statement page of the Step Properties notebook, click **Build SQL**. SQL Assist opens.
2. Click the **Columns** tab.
3. Click **Add** to open the Expression Builder window.

You can use the fields and controls in the Expression Builder to build your expression, or you can type your expression in the **Expression** field.

To build an expression:

1. Use the **Columns**, **Operators**, and **Case** lists to select the components of the expression. Select a particular column, operator, or case keyword to add it to the **Expression** field. Select items in the order that you want them displayed, because each item that you select is appended to the expression in the **Expression** field.
2. Add specific values to your expression. Type a value in the **Value** field, then click the check mark to add the value to the **Expression** field.
3. Add a function to your expression.
4. Add a constant to your expression.
5. Click **OK**. The Expression Builder window closes, and the column expression is added to the **Selected columns** list on the Columns page.
6. Click the **Name** field of the new column, and type the name of the column.
7. Press Enter.
8. Click **Move Up** and **Move Down** to move the column to the appropriate position in the table.

Selecting, inserting, and updating source data in a target table

Related concepts:

- “Star joins” on page 116

Related tasks:

- “Defining a join” on page 112
- “Removing a join” on page 113

Adding a function to an expression in the Expression Builder

You can add functions to expressions that you build in the Expression Builder.

Procedure:

To add a function to your expression:

1. From the Expression Builder window, select a category from the **Functions** list. The list below the **Functions** field changes to display functions in the specified category.
2. Double-click on a function in the list below the **Functions** field.
3. The Function Parameters window opens. Select the format of the function and specify the parameter values.
4. Click **OK**. The Function Parameters window closes. The function and its parameters are displayed in the **Expression** field of the Expression Builder.

Adding a constant to an expression

You can add constants to expressions that you create in the Expression Builder.

Procedure:

To add a constant to your expression:

1. From the Expression Builder window, select a category from the **Constants** list. The list below the **Constants** field changes to display constants in the specified category.
2. Double-click on a constant in the list below the **Constants** field. The constant is added to the expression in the **Expression** field.

Chapter 10. Loading and exporting data

The Data Warehouse Center provides several different ways to move data, depending on your needs. This chapter explains how to use the supplied export and load utilities to move data between files and DB2 and Red Brick Warehouse databases.

Data Warehouse Center load and export programs

You can use the supplied export utilities, such as DB2[®] data export, to extract data from a DB2 Universal Database[™] database and write it to a flat file. You can use the DB2 load append or replace utility, to extract data from a file and write it to another DB2 database on an iSeries[™] system.

Recommendation: Use these utilities instead of the Data Warehouse Center's SQL processing when there is a substantial amount of data to load. Experiment to see which one is more efficient for you.

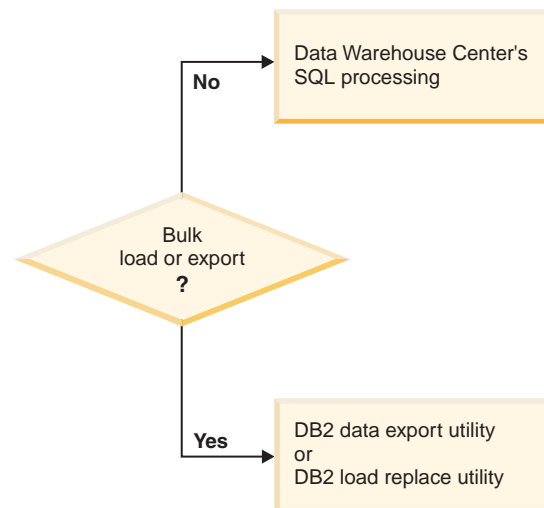


Figure 11. When to use the load and export programs

The bulk load and export utilities operate on a data file and a DB2 database. The database server does not need to reside on the agent site, but the source or target file must reside on the agent site.

These utilities write log files in the directory that is specified by the VWS_LOGGING environment variable. The default value of VWS_LOGGING is `x:\program files\IBM\sqllib\logging\` on Windows[®], and `/var/IWH` on UNIX[®] and z/OS[™], where `x` is the drive on which you installed the warehouse agent.

The sections about the DB2 UDB export and DB2 UDB load warehouse utilities and the Red Brick[™] SQL Export and Red Brick TMU Load warehouse utilities describe how to define the basic values for these utilities.

Related tasks:

- “Defining values for the Red Brick SQL Export utility” on page 123

Loading and exporting data

- “Defining values for the Red Brick TMU Load utility” on page 132

Exporting data

You can use the Data Warehouse Center utility to export data from a DB2 database, a database that is registered in ODBC, or a Red Brick Warehouse database.

Defining values for a DB2 UDB export utility

Use the Properties notebook for the DB2 UDB export utility to define a step that creates DB2 scripts that are run by the warehouse agent. These scripts export data from DB2 tables or views to a file located at the agent site.

The database server does not need to be on the agent site.

The DB2 UDB export utility creates the target file if it does not exist, or replaces it if it exists.

This warehouse utility is available for the following operating systems:

- AIX
- Linux
- Solaris Operating Environment
- Windows

Restrictions:

- The target file must be on the agent site.
- The source tables or views must be linked to the step in the Process Model window.
- The step must be linked to the warehouse target file.
- DB2 UDB export steps do not use the Column Mapping page.
- When you export data from DB2 for z/OS using DB2 UDB export utility, the format must be IXF and the target file type must be Fixed.

Procedure:

To define values for the DB2 UDB export warehouse utility:

1. Link the source table, where you want the data exported from, to the step.
2. Link the target file, where you want the exported data written, to the step.
3. Open the Properties notebook for the step and define the step properties.

Related tasks:

- “Exporting data from tables or views: Data Warehouse Center help”

Defining values for the Data export with ODBC to file utility

Use the Data export with ODBC to file warehouse utility to create a DB2 script that selects data in a table that is contained in a database registered in ODBC, and write the data to a delimited file. To run this utility on AIX or UNIX, use the ODBC version of the warehouse agent.

This utility uses ODBC-accessible (not ODBC) warehouse sources. You can use the target of this utility as a source to other steps.

You connect a source to the step in the Process Model window. The output file is generated on the agent site.

This warehouse utility is available for the following operating systems:

- AIX
- Linux
- Solaris Operating Environment
- Windows

Restrictions:

- The target file must be on the agent site.
- The source tables or views must be linked to the step in the Process Model window.
- The step must be linked to the warehouse target file.

Procedure:

To define values for the Data export with ODBC to file warehouse utility:

1. Link the step to a source that you want to export.
2. Link the step to a file that you want to contain the exported data.
3. Open the Properties notebook for the step, and define the step properties.

Related tasks:

- “Writing data from a database registered in ODBC to a delimited file: Data Warehouse Center help”

Defining values for the Red Brick SQL Export utility

Use the Properties notebook for the Red Brick SQL Export step to define a step that executes an external program that runs the Red Brick SQL Export function. This external program uses the Red Brick SQL Export function to export data from Red Brick tables or views to a file, in either delimited or fixed column format. The exported file is located on the Red Brick Server site.

The database server does not need to be on the agent site, but the ODBC DSN for the database must be defined on the agent site. You do not link warehouse sources or targets to a Red Brick SQL Export step.

The Red Brick SQL Export step creates the output file if it does not exist, and replaces the output file if it does exist. The output file must be on the Red Brick Warehouse server site.

This warehouse utility is available for the following operating systems:

- AIX
- Linux
- Solaris Operating Environment
- Windows

Restrictions:

- For reasons of security, the output file for the export must carry the file extension of .wre. If the given output file does not end with the .wre extension, a .wre extension will automatically be appended.

Loading and exporting data

Procedure:

To define values for the Red Brick SQL Export step:

1. Create a step with the warehouse-supplied Red Brick SQL Export utility. Red Brick SQL Export steps do not use the Column Mapping page.
2. Open the Properties notebook for the step and define the step properties.

Related tasks:

- “Defining values for the Data export with ODBC to file utility” on page 122

Loading data

Use the warehouse load utilities to load data into a DB2 database, including DB2 for iSeries and DB2 for z/OS, or a Red Brick Warehouse database.

Defining values for a DB2 Universal Database load utility

Use the properties notebook for the DB2 Universal Database Load step to create a step that creates a DB2 script that loads data from a source or target file into a DB2 Universal Database table. The scripts are run by the warehouse agent. You can use this step to load data into multi-partition target tables.

You can use a warehouse source or target file as a source file for this step. Link the source to the step in the Process Model window. Then, link the step to a warehouse target, or specify that the Data Warehouse Center will create the target table.

This warehouse utility is available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Linux
- Solaris Operating Environment

Restrictions:

- The Column Mapping page is not available for this step.
- The step must be linked to a source and target file.

Procedure:

To define values for a DB2 UDB Load step:

1. Define a flat-file warehouse source for your source file. In the **File name** field, type the fully qualified file name.
2. Create a step with the warehouse-supplied DB2 UDB Load utility.
3. Select your flat-file source, and add the source file to the step.
4. Select your target table from warehouse target and connect with the step.
5. Promote the step to test mode and run it. The target table now contains all the source data from your flat file.

Defining a DB2 for iSeries Data Load Insert utility

The DB2 for iSeries Data Load Insert utility uses programs to load data from a flat file to a DB2 UDB for iSeries table. The load operation appends new data to the end of existing data in the table.

Before you define this step, you must connect the step to a warehouse source and a warehouse target in the Process Modeler.

Acceptable source files are iSeries QSYS source file members or stream files in Integrated File System (IFS), the root file system.

Tip: You can improve both performance and storage use by using QSYS file members instead of stream files. CPYFRMIMPF makes a copy of the entire stream file to QRESTORE and then loads the copy into your table. See the online help for CPYFRMIMPF for more information.

Target tables can reside only in the QSYS file system.

You can make changes to the step only when the step is in development mode.

Before the step loads new data into the table, it exports the table to a backup file, which you can use for recovery.

The Column Mapping page is not available for this step.

Prerequisites:

The user profile under which this utility and the warehouse agent run must have at least read/write authority to the table that is to be loaded.

The following requirements apply to the DB2 for iSeries Load Insert utility. For information about the limitations of the CPYFRMIMPF command, see the restrictions section of the online help for the CPYFRMIMPF command. To view the online help for this command, type CPYFRMIMPF on the iSeries command prompt, and press F1.

1. The Data Warehouse Center definition for the agent site that is running the utility must include a user ID and password. The database server does not need to be on the agent site. However, the source file must be on the database server. Specify the fully qualified name of the source files as defined on the DB2 server system.
2. If the load process generates warnings, the utility returns as successfully completed.
3. The default behavior for the DB2 for iSeries Load Insert utility is to tolerate all recoverable data errors during LOAD (ERRLVL(*NOMAX)).

To override this behavior, include the ERRLVL(n) keyword in the fileMod string parameter, where n = the number of permitted recoverable errors.

You can find more information about the ERRLVL keyword in the online help for the CPYFRMIMPF command.

Procedure:

To define properties for a DB2 for iSeries Data Load Insert step:

Loading and exporting data

1. Define a flat-file warehouse source for your source file. In the **File name** field, type the fully qualified file name.
2. Create a step with the warehouse-supplied DB2 for iSeries Load Insert utility.
3. Select your flat-file source, and add the source file to the step.
4. Select your target table from warehouse target and connect with the step.
5. Promote the step to test mode and run it. The target table now contains all the source data from your flat file.

Related tasks:

- “Defining values for a DB2 Universal Database load utility” on page 124
- “Defining a DB2 for z/OS load utility” on page 130
- “Defining values for the Red Brick TMU Load utility” on page 132

Defining a DB2 for iSeries Data Load Replace utility

The DB2 for iSeries Data Load Replace utility uses programs to load data from a flat file to a DB2 UDB for iSeries table. The load operation completely replaces existing data in the table.

Acceptable source files for the iSeries implementation of the DB2 for iSeries Data Load Replace utility are iSeries QSYS source file members or stream files in Integrated File System (IFS), the root file system.

Tip: You can improve both performance and storage use by using QSYS file members instead of stream files. CPYFRMIMPF copies the entire stream file to QRESTORE and then loads the copy into your table.

The following behaviors apply to the DB2 for iSeries Data Load Replace utility for iSeries:

- If the utility detects a failure during processing, the table will be emptied. If the load generates warnings, the utility returns as successfully completed.
- This implementation of the DB2 for iSeries Data Load Replace utility differs from load utilities on other platforms. Specifically, it does not delete all loaded records if the load operation fails for some reason.

Normally, this utility replaces everything in the target table each time it is run, and automatically deletes records from a failed run. However, if the load operation fails, avoid using data in the target table. If there is data in the target table, it will not be complete.

- The default behavior for the DB2 for iSeries Data Load Replace utility is to tolerate all recoverable data errors during LOAD (ERRLVL(*NOMAX)).

To override this behavior, include the ERRLVL(n) keyword in the fileMod string parameter, where n = the number of permitted recoverable errors.

You can find more information about the ERRLVL keyword in the online help for the CPYFRMIMPF command.

Prerequisites:

- Before you define this step, connect the step to a warehouse source and a warehouse target in the Process Modeler.

Restrictions:

- Target tables can reside only in the QSYS file system.
- You can make changes to the step only when the step is in development mode.

- The Column Mapping page is not available for this step.
- The user profile under which this utility and the warehouse agent run must have at least read/write authority on the table that is to be loaded.
- The Data Warehouse Center definition for the agent site that is running the utility must include a user ID and password. The database server does not need to be on the agent site. However, the source file must be on the database server. Specify the fully qualified name of the source files as defined on the DB2 server.
- You cannot use SQL statements on flat files. SQL is not supported in the current version of the iSeries warehouse agent. You also cannot use sample contents of a flat file using the iSeries warehouse agent. The data must be transferred using the iSeries Load with Replace warehouse utility.

Procedure:

To define a process to load a local file into a local table:

This process will start the DB2 for iSeries Load Replace utility and load the local table with the local file.

1. Define a flat-file warehouse source for your source file. In the **File name** field, type the fully qualified file name.
2. Create a step with the warehouse-supplied DB2 for iSeries Load Replace utility.
3. Select your flat-file source, and add the source file to the step.
4. Select your target table from warehouse target and connect with the step.
5. Promote the step to test mode and run it. The target table now contains all the source data from your flat file.

Related tasks:

- “Loading data from a flat file into a DB2 UDB for AS/400 table, replacing existing data: Data Warehouse Center help”

Modstring parameters for DB2 for iSeries load utilities

You can use the Modstring parameters to modify the file characteristics that the CPYFRMIMPF command expects the input file to have. If this parameter is omitted, all of the default values that the CPYFRMIMPF command expects are assumed to be correct.

Some of the default characteristics of the input file are:

- The file is comma delimited.
- Strings and date/time values are enclosed in quotation marks.
- Date and time values are in ISO format.
- The decimal point is represented by a period character.

For more information about the default values for the CPYFRMIMPF command, see the iSeries™ online help for the CPYFRMIMPF command.

The format for the FileMod string is:

- The string must contain valid CPYFRMIMPF command keywords. All of the valid keywords for the CPYFRMIMPF command are described in the online help for the command.
- Each keyword must be followed immediately by its value. The value must be enclosed in parentheses.

Loading and exporting data

- Each keyword must be separated from the next keyword by a space.

Certain parameters require that you enclose values in single quotation marks. For example, the FLDDL command must have the values enclosed by single quotation marks. The Data Warehouse Center generates the iSeries CALL statement for VWPLOADI in the form:

```
CALL PGM(QIWH/VWPLOADI) PARM('fromfile' 'totable' 'filemodstring')
```

Two single quotation marks together tell the iSeries command prompt processor that your parameter value contains a single quotation mark. This behavior prevents the command line processor from interpreting a single quotation mark as the end-of-parameter marker.

Trace files for the DB2 for iSeries load utilities

The DB2® for iSeries™ Load Insert and DB2 for iSeries Load Replace utilities provide two kinds of diagnostic information:

- The return code, as documented in the Data Warehouse Center online help
- The VWPLOADI trace for the DB2 for iSeries Load Insert utility, and the VWPLOADR trace for the DB2 for iSeries Load Replace utility.

Successful completion of these utilities does not guarantee that the data was transferred correctly. For stricter error handling, use the ERRLVL parameter.

Reading the VWPLOADI and the VWPLOADR trace file

The VWPLOADI and VWPLOADR trace files are located in the Integrated File System in the /QIBM/UserData/IWH directory.

The VWPLOADI and VWPLOADR trace file has the following name format:

```
VWxxxxxxxx.programName
```

xxxxxxxx

The process ID of the VWPLOADI run that produced the file.

programName

The name of the program, VWPLOADI or VWPLOADR.

where:

Viewing the VWPLOADI and VWPLOADR trace with Client Access/400

You should now be able to load the file into any ASCII text editor or word processor.

iSeries exceptions

If there was a failure of any of the system commands issued by the DB2 for iSeries Load Insert or DB2 for iSeries Load Replace utility, then there will be an exception code recorded in the trace file. To get an explanation for the exception:

1. At an iSeries command prompt, enter DSPMSGD RANGE(XXXXXXX), where XXXXXXX is the exception code. For example, you might enter DSPMSGD RANGE(CPF2817).

The Display Formatted Message Text window is displayed.

2. Select option **30** to display all information.

The second line in the trace file contains the information that you need to issue the WRKJOB command.

To view the spool file, you can copy and paste the name of the message file to an iSeries command prompt after the WRKJOB command and press Enter. View the spool file for the job to get additional information about any errors that occurred.

Viewing trace files for the DB2 for iSeries load utilities

You can view VWPLOADI and VWPLOADR trace files from a workstation and with Client Access/400.

Prerequisites:

If you use Client Access/400 to access the trace file, you must define to Client Access/400 the file extension that is appropriate for the program.

Procedure:

To view VWPLOADI or VWPLOADR trace files from a workstation:

1. Use Client Access/400 to map your iSeries root file system to a logical drive, or use FTP to copy the file to the workstation.
2. Open the trace file with a text editor to view the information.

To use Client Access/400 to map an iSeries system to a logical drive on a Windows workstation:

1. Set up a Client Access/400 connection to your iSeries system over TCP/IP.
2. Open the Windows Explorer program for your operating system.
3. From the Explorer menu, click **Tools** → **Map Network Drive**.
4. Type the path name:
\\hostname\.

where hostname is the fully qualified TCP/IP host name of your iSeries system.

5. Click **OK**.

Related tasks:

- “Defining a file extension to Client Access/400” on page 129

Defining a file extension to Client Access/400

If you use Client Access/400 to access the trace file, you must define the file extension that is appropriate for the utility to Client Access/400. (For example, .VWPLOADI for the DB2 for iSeries utility and .VWPLOADR for the DB2 for iSeries utility.) Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

Procedure:

To define a file extension to Client Access/400:

1. From Windows, click **Start** → **Programs** → **IBM AS400 Client Access Express** → **AS/400 Operation Navigator**.
The Operation Navigator notebook opens.
2. Expand the system name node in the Operation Navigator.
3. Expand the File Systems tree.
4. Right-click **Integrated File System**, and click **Properties**.

Loading and exporting data

5. Select all file extensions.
6. Click **Add**.
7. Click **OK**.
8. Load the file into any ASCII text editor or word processor.

Related concepts:

- “Modstring parameters for DB2 for iSeries load utilities” on page 127
- “Trace files for the DB2 for iSeries load utilities” on page 128

Related tasks:

- “Defining values for a DB2 Universal Database load utility” on page 124
- “Viewing trace files for the DB2 for iSeries load utilities” on page 129
- “Defining a DB2 for z/OS load utility” on page 130
- “Defining values for the Red Brick TMU Load utility” on page 132

Defining a DB2 for z/OS load utility

The DB2 for z/OS load utility uses DSNUTILS to load records into one or more tables in a table space.

When you define values for DB2 for z/OS load utility, the following information applies:

DB2 unload format

The DB2 unload format specifies that the input record format is compatible with the DB2 unload format. The DB2 unload format is the result of REORG with the UNLOAD ONLY option. Input records that were uploaded by the REORG utility are loaded into the tables from which they were unloaded. Do not add or change the column specifications between the REORG UNLOAD ONLY and the LOAD FORMAT UNLOAD. DB2 reloads the records into the same tables from which they were unloaded.

SQL/DS unload format

The SQL/DS unload format specifies that the input record format is compatible with the SQL/DS unload format. The data type of a column in the table to be loaded must be the same as the data type of the corresponding column in the SQL/DS table. SQL/DS strings that are longer than the DB2 limit cannot be loaded.

Specify resume option at table space level

Click **NO** to load records into an empty table space. If the table space is not empty, and you did not specify **REPLACE**, the LOAD process ends with a warning message. For nonsegmented table spaces that contain deleted rows or rows of dropped tables, using the **REPLACE** option provides more efficiency.

Specify resume option at table space level

Click **YES** to drain the table space, which can inhibit concurrent processing of separate partitions. If the table space is empty, a warning message is issued, but the table space is loaded. Loading begins at the current end of data in the table space. Space occupied by rows marked as deleted or by rows of dropped tables is not reused.

Procedure:

To define values for a step that uses this utility:

1. Define a warehouse target where the DSNUTILS stored procedure will run.
2. Create a step that uses DB2 for z/OS load utility.
3. Connect the step to the target.
4. Open the step notebook, and define values for the step.

Related tasks:

- “Loading data into tables of a DB2 for OS/390 table space: Data Warehouse Center help”

Copying data between DB2 utilities

When you want to copy a table by unloading it into a flat file, then loading the flat file to a different table, you normally have to unload the data, edit the load control statements that unload produces, then load the data. Using the zSeries warehouse agent, you can specify that you want to reload data to a different table without stopping between steps and manually edit the control statements.

Procedure:

To copy data between DB2 for z/OS and OS/390 tables using the LOAD utility:

1. Use the utility panel to create a step that unloads a file using the UNLOAD utility or the REORG TABLESPACE utility. Both of these utilities produce two output data sets, one with the table data and one with the utility control statement that can be added to the LOAD utility.

This is an example of the DSNUTILS parameters you might use for the Reorg Unload step:

```
UTILITY_ID REORGULX
RESTART NO
UTSTMT REORG TABLESPACE DBVW.USAINENT UNLOAD EXTERNAL
UTILITY_NAME REORG TABLESPACE
RECDSN DBVW.DSNURELD.RECDSN
RECDEVT SYSDA
RECSpace 50
PNCHDSN DBVW.DSNURELD.PNCHDSN
PNCHDEVT SYSDA
PNCHSPACE 3
```

2. Use the utility panel to create a load step. The DSNUTILS utility statement parameter specifies a utility control statement. The warehouse utility interface allows a file name in the **Utility statement** field. You can specify the file that contains the valid control statement using the keyword :FILE:, and the name of the table that you want to load using the keyword :TABLE:.
3. To use the LOAD utility to work with the output from the previous example, apply the following parameter values in the LOAD properties:

```
UTILITY_ID LOAD
RESTART NO
UTSTMT :FILE:DBVW.DSNURELD.PNCHDSN:TABLE:[DBVW].INVENTORY
UTILITY_NAME LOAD
RECDSN DBVW.DSNURELD.RECDSN
RECDEVT SYSDA
```

4. In the **UTSTMT** field, type either a load statement or the name of the file that was produced from the REORG utility with the UNLOAD EXTERNAL option. The previous example will work for any DB2 for z/OS and OS/390 source

Loading and exporting data

table or target table, whether these tables are on the same or different DB2 subsystems. The control statement flat file can be either HFS or native OS/390 files.

Related tasks:

- “Defining a DB2 for z/OS load utility” on page 130

Defining values for the Red Brick TMU Load utility

Use the Properties notebook for the Red Brick TMU Load step to define a step that executes an external program that runs the Red Brick TMU Load utility. This external program uses the Red Brick TMU Load utility to load data from an external file to a Red Brick table. You can use three different versions of the TMU Load utility programs: Serial TMU Load, Parallel TMU Load, and Remote TMU Load. The external program is run by the warehouse agent. You do not link warehouse sources or targets to a Red Brick SQL Load step.

The file to be loaded must be located on the warehouse agent site.

For the Remote TMU Load utility, the database server does not need to be located on the warehouse agent site; however, a ODBC DSN must be defined on the agent site. For the Serial TMU Load and Parallel TMU Load utilities, the Red Brick database server must be located on the agent site.

The Red Brick TMU Load utility is available for the following operating systems:

- AIX
- Linux
- Solaris Operating Environment
- Windows

Restrictions:

- The file to be loaded must be located on the agent site.

Procedure:

To define values for the Red Brick TMU Load step:

1. Create a step with the warehouse-supplied Red Brick TMU Load utility. You must provide predefined TMU Load scripts to this step as a part of the input parameters.
2. Open the Properties notebook for the step and define the step properties. Red Brick TMU Load steps do not use the Column Mapping page.

Related tasks:

- “Accessing data files with the Copy file using FTP warehouse program” on page 65
- “Manipulating files using FTP or the Submit JCL jobstream warehouse program” on page 133

Chapter 11. Moving files and tables

This chapter describes how to define the different step subtypes for the warehouse file programs, and how to set up replication in the Data Warehouse Center.

Manipulating files using FTP or the Submit JCL jobstream warehouse program

Use the following programs to manipulate files:

Copy file using FTP warehouse program

Use the warehouse program Copy file using FTP to copy files on the agent site to and from a remote host. When you define a step that uses this warehouse program, select a source file and select a target file.

Run FTP command file warehouse program

Use the warehouse program Run FTP command file to transfer files from a remote host using FTP. When you define a step that uses this warehouse program, do not specify a source or target table for the step. If you want to load a remote flat file into a table on an iSeries system, you can combine this step with an iSeries load step. The FTP step transfers the remote flat file to a local file on the iSeries system, and then the load step loads the data into a table.

Submit z/OS JCL jobstream warehouse program

Use the Submit z/OS JCL jobstream warehouse program to submit a JCL jobstream that resides on z/OS to an z/OS system for execution. The Submit z/OS JCL jobstream warehouse program also creates a JES log file at the agent site. It erases the copy of the JES log file from any previous jobs on the agent site before submitting a new job for processing. It also verifies that the JES log file is downloaded to the agent site after the job completes. When you define a step that uses this warehouse program, do not specify a source or target table for the step. This warehouse program runs successfully if the z/OS host name, user ID, and password are correct.

The Copy file using FTP and Run FTP command file programs are available for the following operating systems:

- AIX
- iSeries
- Linux
- Solaris Operating Environment
- Windows
- z/OS

The Submit JCL jobstream program is also available for all of the operating systems mentioned above except for iSeries.

Prerequisites:

You must meet the following prerequisites before using these programs:

Copy file using FTP warehouse program

Before you copy files to z/OS using the Copy file using FTP program, you

Moving files and tables

must allocate their data sets. One file must be stored on the agent site, and the other must be stored on the z/OS system.

Submit z/OS JCL jobstream warehouse program

- Before you use the Submit z/OS JCL jobstream warehouse program, test your JCL file by running it from TSO under the same user ID that you plan to use with the program.
- The Submit z/OS JCL jobstream warehouse program requires TCP/IP 3.2 or later installed on z/OS. Verify that the FTP service is enabled before using the program.

Restrictions:

The following restrictions apply when you are using these programs:

Copy file using FTP warehouse program

You cannot transfer VSAM data sets using the Copy file using FTP program.

Submit z/OS JCL jobstream program

- Column mapping is not available for this step.
- The job must have a MSGCLASS and SYSOUT routed to a held output class.

Procedure:

To manipulate files using warehouse programs, open the Properties notebook for the step, and specify properties for the step.

Accessing a remote file on a secure UNIX or UNIX System Services system

If you are having problems accessing a remote file on a secure UNIX[®] or UNIX Systems Services system, verify that you have write access to the \$HOME directory of the user specified in the agent site definition.

For example, the host name of the agent site is coffee.stl.ibm.com. You want to transfer a file using FTP from the remote site morning.stl.ibm.com to the agent site, using the remote user ID vwinst2. The ~vwinst2/.netrc file must contain the following entry:

```
machine coffee.stl.ibm.com login vwinst2
```

Related tasks:

- “Importing table definitions from a Microsoft Access database (Windows)” on page 51

Replication

This section describes how to set up replication in the Data Warehouse Center.

Replication is the process of maintaining a defined set of data in more than one location. Replication involves copying designated changes from one location (a source) to another (a target), and synchronizing the data in both locations. The source and target can be in logical servers (such as a DB2[®] database or a DB2 for

z/OS™ subsystem or data-sharing group) that are on the same computer or on different computers in a distributed network.

Replication in the Data Warehouse Center

You can use the replication capabilities of the Data Warehouse Center when you want to keep a warehouse table synchronized with an operational table without fully loading the table each time that the operational table is updated. With replication, you can use incremental updates to keep your data current.

You can use the Data Warehouse Center to define a replication step, which will replicate changes between any DB2 relational databases. You can also use other IBM® products (such as DB2 Information Integrator and IMS™ DataPropagator™ NonRelational) or non-IBM products (such as Microsoft® SQL Server and Sybase SQL Server) to replicate data between many database products, both relational and nonrelational. The replication environment that you need depends on when you want to update data and how you want to handle transactions.

To define a replication step with the Data Warehouse Center, you must belong to a warehouse group that has access to the process in which the step will be used.

The Data Warehouse Center supports five types of replication:

User copy

A complete, condensed copy of the replication source table. Complete means that the target table contains one row for every row in the source table. Condensed means that the target table has a primary key and that updates to the target table are made using the primary key. User Copy tables look like regular source tables and are a good starting point for replication. They are the most common type of target table.

Point-in-time

A complete, condensed copy of the replication source table at a certain time. The target table has a primary key and a timestamp column. The timestamp column indicates when a transaction occurred. Use this type of table if you want to keep track of transaction times.

Base aggregate

A history table in which new rows are appended for each subscription cycle using the result of an SQL column function calculation on the replication source table (the Base table).

Change aggregate

A history table in which new rows are appended for each subscription cycle using the result of an SQL column function calculation against the replication source change-data table, which contains the recently changed data.

Staging table

A table that produces target tables that contain data from committed transactions. Also called consistent change-data tables, these tables contain different types of data if they are condensed, noncondensed, complete, or noncomplete.

- A *condensed* staging table has a primary key and contains only the most current value for a row. The Apply program makes changes using the primary key. Condensed tables are useful for staging changes to remote locations and for summarizing hot-spot changes before they are replicated to targets.

Moving files and tables

- A *noncondensed* staging table does not have a primary key, and it contains a history of changes to a row. The Apply program appends rows to noncondensed tables. Noncondensed staging tables are useful for auditing purposes.
- A *complete* staging table contains one row for every row in the source table.
- A *noncomplete* staging table is empty when it is created, and rows are appended as changes are made to the source table for insert, update, and delete operations.

When you promote a replication step to test mode, the target table is created and the subscription set in the replication control table is enabled. The first time a replication step is run, a full refresh copy is made. Promoting a replication step to production mode will enable the schedules that are defined. You can make changes to a step only when it is in development mode.

Related concepts:

- “Replication source tables in the Data Warehouse Center” on page 136
- “Replication password files” on page 140

Related tasks:

- “Setting up replication in the Data Warehouse Center” on page 137
- “Creating replication control tables” on page 138
- “Defining a replication step in the Data Warehouse Center” on page 139
- “Using a replication step in a process” on page 139

Related reference:

- “Replication programs” on page 93

Replication source tables in the Data Warehouse Center

In the Data Warehouse Center, you define replication sources in the same way that you define other relational sources. A table or view must be defined as a replication source using the DB2[®] Replication Center before it can be used as a replication source in the Data Warehouse Center. When you import a replication source, be sure to select the check box to retrieve replication source tables and specify the correct Capture schema. The Data Warehouse Center reads the available replication source tables from the IBMSNAP_REGISTER table using the specified Capture schema. The default capture schema is ASN.

When you define a replication source table in DB2 Replication Center, you must choose which before-image and after-image columns to replicate. The before-image and after-image columns are then defined in the replication change-data table, which is created by the DB2 Replication Center. In the change-data table, the before-image column names start with a special character (usually an X). When a replication source is imported into the Data Warehouse Center, the defined before and after-image columns will be available.

If you need to change the before and after-image columns that are enabled for replication, you must redefine the replication source in the DB2 Replication Center. Then you must import the replication source again into the Data Warehouse Center.

Related concepts:

- “Replication in the Data Warehouse Center” on page 135
- “Replication password files” on page 140

Related tasks:

- “Setting up replication in the Data Warehouse Center” on page 137
- “Creating replication control tables” on page 138
- “Defining a replication step in the Data Warehouse Center” on page 139
- “Using a replication step in a process” on page 139

Related reference:

- “Replication programs” on page 93

Setting up replication in the Data Warehouse Center

This topic describes how to set up replication in the Data Warehouse Center.

Prerequisites:

- Replication control tables must exist in the warehouse control database and the warehouse target database.
- The Data Warehouse Center includes a JCL template for replication support. If you plan to use the zSeries warehouse agent to run the Apply program, change the account and data set information in this template for your z/OS system.
- To use the warehouse agent for replication, if the source, target, capture control, or apply control servers (databases) are remote to the client system, you must catalog the database with the same name, user ID, and password on both the client and the warehouse agent systems. After you catalog the source on the client and warehouse agent systems, verify that you can connect to the source, target, capture, and apply databases.

If you cannot connect to the warehouse source, warehouse target, replication capture, or replication apply databases, verify that the environment variable DB2COMM on the remote system is set to TCP/IP, and that the port number matches the port number of the node that is cataloged on the client system. To verify the port number on the remote system, enter the following command on a DB2 command line:

```
get dbm cfg | grep SVCENAME
```

You specify the port number of the client system when you catalog the node.

Procedure:

To set up replication in the Data Warehouse Center:

1. Create the replication control tables by using the Replication Center.
2. Register a source table by using the Replication Center.
3. Import the defined replication source table into the Data Warehouse Center.
4. Define a replication step in the Data Warehouse Center using the replication source table as input.
5. Run the replication password utility program `asnpwd` to create the password file that is required by your warehouse step. Replication steps assume that the password file for the step is in the `VWS_LOGGING` (environment variable) directory, and has a file name of `applyqual.pwd` where `applyqual` is the apply

Moving files and tables

qualifier for the step. This information is added to Apply program command before the Apply program is started by the warehouse agent.

6. Start the Capture program on the same system as the source database.
7. Promote the replication step to test mode or production mode in the Data Warehouse Center. When the step is promoted, the target table is created and the replication subscription is written to the replication control tables.
8. Run the step. When the step runs, the warehouse agent starts the Apply program to process the replication subscription.

Related concepts:

- “Replication in the Data Warehouse Center” on page 135
- “Replication source tables in the Data Warehouse Center” on page 136
- “Replication password files” on page 140

Related tasks:

- “Using the z/OS warehouse agent to automate replication apply steps” in the *DB2 Warehouse Manager Standard Edition Installation Guide*
- “Creating replication control tables” on page 138
- “Defining a replication step in the Data Warehouse Center” on page 139
- “Using a replication step in a process” on page 139

Related reference:

- “Replication programs” on page 93

Creating replication control tables

Replication control tables must exist in the warehouse control database and the target database before you can set up replication in the Data Warehouse Center. The replication control tables are in the ASN schema. The control table names start with IBMSNAP. If the control tables do not exist, you can create them in the Replication Control Center.

Procedure:

To create the control tables:

1. Open the Replication Center.
2. Expand the **Replication Definitions** folder, and right-click the **Capture Control Servers** folder.
3. Click **Create Capture Control Tables -> Quick**.
4. In the Select a Server window, select the warehouse control database, and click **OK**.
5. Specify the information in the Create Control Tables Quick window, or accept the default values for all settings.
6. Click **Finish**.
7. Close the message window.
8. In the Run Now or Save SQL window, remove the SQL statements that insert into the IBMSNAP_REGISTER table and the statements that create the change-data table.
9. Click **OK** to run the SQL script.

10. Repeat these steps to create the replication control tables in the warehouse target database.

Related concepts:

- “Replication in the Data Warehouse Center” on page 135
- “Replication source tables in the Data Warehouse Center” on page 136
- “Replication password files” on page 140

Related tasks:

- “Setting up replication in the Data Warehouse Center” on page 137
- “Defining a replication step in the Data Warehouse Center” on page 139
- “Using a replication step in a process” on page 139

Related reference:

- “Replication programs” on page 93

Defining a replication step in the Data Warehouse Center

The Data Warehouse Center supports five types of replication.

Procedure:

To define a replication step, open the Properties notebook for the step that you want to define, and specify the step properties.

Related concepts:

- “Replication in the Data Warehouse Center” on page 135

Related tasks:

- “Creating replication control tables” on page 138
- “Defining a base aggregate replication step: Data Warehouse Center help”
- “Defining a change aggregate replication step: Data Warehouse Center help”
- “Defining a point-in-time replication step: Data Warehouse Center help”
- “Defining a staging table replication step: Data Warehouse Center help”

Using a replication step in a process

This topic explains how to use a replication step in a Data Warehouse Center process.

Prerequisites:

Before you can run a replication step in a process, you must create the password file for the replication step and its Apply qualifier.

Procedure:

To use a replication step in a process:

1. Define and open a process in the Data Warehouse Center.
2. Add a replication source table to the process.
3. Add a replication step to the process.

Moving files and tables

4. Link the source table to the replication step that you are using.
5. Define the properties of the replication step.
6. Start the Capture program.

To start the Capture program, enter the following command on a Windows command prompt:

```
ASNCCP CAPTURE_SERVER=source-database STARTMODE=WARMSI
```

The STARTMODE=WARMSI parameter specifies an initial cold start that deletes any existing data in the change-data tables; subsequent starts are warm starts. You can also start the Capture program from the Replication Center. Leave the Capture program running and start the Apply program.

7. Test the step.

When the step runs, the event record in the IBMSNAP_SUBS_EVENT table is updated, the subscription record in IBMSNAP_SUBS_SET is marked active, and the warehouse agent starts the Apply program.

8. Schedule the step.
9. Promote the step to production mode.

Related concepts:

- “Replication in the Data Warehouse Center” on page 135
- “Replication source tables in the Data Warehouse Center” on page 136
- “Replication password files” on page 140

Related tasks:

- “Setting up replication in the Data Warehouse Center” on page 137
- “Creating replication control tables” on page 138
- “Defining a replication step in the Data Warehouse Center” on page 139

Related reference:

- “Replication programs” on page 93

Replication password files

Depending on the version of the Data Warehouse Center that you use, either you must set up and maintain the replication password files, or the warehouse agent creates them for you.

Replication password files in Version 8 or later

For replication in the Data Warehouse Center, Version 8 or later, you must set up and maintain the replication password files by using the replication program `asnpwd`. Replication password files are encrypted.

The Data Warehouse Center replication steps assume that the password file for a replication step are in the environment variable `VWS_LOGGING` directory and have a file name of `applyqual.pwd`, where `applyqual` is the Apply qualifier for the Data Warehouse Center replication step. This information is added to Apply program command before the Apply program is started by the Data Warehouse Center agent. The following example shows an Apply program command:

```
DW Step Apply Qualifier = dwqual  
VWS_LOGGING = c:\sqllib\logging  
Control Server = replserv
```


Produces the following Apply command:

```
asnapply apply_qual=dwqual control_server=replserv  
copyonce=y trlreuse=y apply_path=c:\sql11lib\logging pwdfile=dwqual.pwd
```

Replication password files in Version 7

In the Data Warehouse Center, Version 7, the warehouse agent creates the replication password file before you run the warehouse replication step. When the step finishes, the password file is deleted. Unlike the Data Warehouse Center, Version 8, you do not have to maintain the replication password files. The password files are not encrypted.

The password file name is *applyqual.pwd*, where *applyqual* is the Apply qualifier for the step. If you specify *ASNPATH*, the password file is created in the environment variable *ASNPATH* directory. If you do not specify *ASNPATH*, the password file is created in the current directory on the system where the Apply program runs.

Related concepts:

- “Replication in the Data Warehouse Center” on page 135
- “Replication source tables in the Data Warehouse Center” on page 136

Related tasks:

- “Setting up replication in the Data Warehouse Center” on page 137
- “Creating replication control tables” on page 138
- “Defining a replication step in the Data Warehouse Center” on page 139
- “Using a replication step in a process” on page 139

Related reference:

- “Replication programs” on page 93

Moving files and tables

Chapter 12. Transforming data

This chapter describes the methods of transforming data that the Data Warehouse Center provides. The chapter describes how to use warehouse transformers to transform data. It also describes how to use Trillium software with the Data Warehouse Center to clean name and address data.

Using transformers in the Data Warehouse Center

The following topics describe how to use warehouse transformers to transform target tables.

Transforming target tables

Use warehouse transformers to perform the following basic data transformations:

- Cleaning data
- Generating key columns
- Generating period tables
- Inverting data
- Pivoting data

Prerequisites:

You must create the appropriate rules table for your clean type before you can use the Clean Data transformer. Not all clean types require a rules table. A rules table designates the values that the Clean Data transformer uses at runtime to clean the source data. The rules table must be in the same database as the source table and target table.

Procedure:

To transform a target table, define the transformer step that you want to use for the transformation.

Related tasks:

- “Cleaning data: Data Warehouse Center help”
- “Assigning key column values: Data Warehouse Center help”
- “Creating a period table: Data Warehouse Center help”
- “Inverting rows and columns in a table: Data Warehouse Center help”
- “Grouping related data from selected columns into a single column: Data Warehouse Center help”

Clean Data transformer

You can use the Clean Data transformer to perform basic cleaning, substitution, and mapping operations on source data. The Clean Data transformer operates on specified data columns of the source table that your step accesses. Then the transformer inserts new rows in the target table to which your step writes. Depending on the processing options that you select, data that cannot be cleaned

Transforming data

will be written to the target error table. You can also use the Clean Data transformer to clean and standardize data values after loading or importing data as part of a process.

The Data Warehouse Center z/OS™ warehouse agent supports the Clean Data transformer only when you run interpreted stored procedures, which requires DB2® Universal Database for z/OS Version 7 and later. In addition, the Encode invalid values clean type requires DB2 Universal Database™ for z/OS Version 8. If you specify the Encode invalid values clean type before installing the DB2 Universal Database for z/OS Version 8, you will receive the return code SQLcode: -443, or the message DWCC-104.

Clean types for the Clean Data transformer

The Clean Data transformer provides the following clean types that you can specify for relational source data:

Find and replace

Locates the selected source column value in the Find column of the rules table, then replaces the value in the target table with the corresponding replacement value in the rules table. A rules table is required for this clean type. A rules table designates the values that the Clean Data transformer will use during the find and replace process. The following find-and-replace types are available:

- Find exact matches
- Find whole string, replace whole string (Available only for CHARACTER data types)
- Find substring, replace whole string (Available only for CHARACTER data types)
- Find substring, replace substring (Available only for CHARACTER data types)

Use the Matching Options window to specify how you want matches to be handled.

When a match is found, the corresponding replacement value is written to the target table. If you specify a differentiator column for both the rules and the source table, then the values in the differentiator column must match for the match to be successful. If you specify an order column for the rules table, the find-and-replace operation matching order follows the ascending order of values in the order column.

If a match is not found, and error processing is enabled, the entire input row is written to the error table with the RUN_ID of the operation.

If you allow nulls for this clean type, you must put a null value in the Find column of the rules table.

Numeric™ clip

Shortens numeric input values that are not within the specified range. Input values within the range are written to the output without any changes. Input values outside the range are replaced by the lower bound replacement value, or the upper bound replacement value. A rules table is required for this clean type. Use the Matching Options window to specify how you want matches to be handled. For numeric values only.

Discretize into ranges

Performs discretization of input values based on the ranges in the rules table. A rules table is required for this clean type.

If you allow nulls for this clean type, you must put a null value in the Bound column of the rules table. Use the Matching Options window to specify how you want matches to be handled.

Carry over with null handling

Specifies columns in the input table to copy to the output table. You can select multiple columns from the input table to carry over to the output table. A rules table is not required for this clean type. This clean type allows you to replace null values with a specified value. You can also reject nulls and write the rejected rows to the error table. Use the Matching Options window to specify how you want matches to be handled.

Convert case

Converts the characters in the source column from uppercase to lowercase or from lowercase to uppercase, and inserts them into the target column. The default is to convert the characters in the source column to uppercase. A rules table is not required for this clean type. Use the Matching Options window to specify how you want matches to be handled.

Encode invalid values

Replaces with the specified value any values that are not contained in the valid values column of the rules table that you are using. You specify the replacement value in the Properties notebook for the Clean Data transformer. You must specify a replacement value that is of the same data type as the source column. For example, if the source column is of type numeric, then you must specify a numeric replacement value. Valid values are not changed when they are written to the target table. A rules table is required for this clean type.

To use the Encode invalid values clean type for a Data Warehouse Center z/OS warehouse agent, you must install the DB2 Universal Database for z/OS Version 8. If you specify this clean type before installing the DB2 Universal Database for z/OS Version 8, you will receive the return code SQLcode: -443, or the message DWQC-104.

Additional functions

The Clean Data transformer also provides you with the following features to be used with the clean data rules.

Data types support: The following table shows the data types that are supported for each clean type:

Table 24. Data types support

Clean type	Character	Numeric	Date/Time	Graphic	LOB
Carry over with null handling	✓	✓	✓	✓	✓
Find and replace	✓	✓	✓		
Discretize into ranges	✓	✓	✓		
Numeric clip		✓			
Convert case	✓				
Encode invalid values	✓	✓	✓		

The carry over clean type supports Graphic and LOB data types only when Insert and Select SQL statements allow this type of column.

Transforming data

Null support: You can explicitly specify whether to allow null values for cleansing. Select the Allow Nulls check box on the Parameters page of the Properties notebook for the Clean Data transformer to treat null values like any other source value. You can replace null values by specifying a replacement value in the Null Replace Value field. You must specify a replacement value that is compatible with the data type of the source column. In clean types that use a rules table, the replacement value must be compatible with the data type of the rules replace column. The replacement value overrides the rules for null value handling in the rules table.

If you do not allow nulls in the transform, the null record is written to an error table, regardless of whether the rules handle null values.

Matching: The Clean Data transformer provides the following matching options:

Create all matches

This is the default option. You can create all the matching values into the target table. This means a 1:n mapping between an input row and its corresponding output rows. For example, this might happen when you have duplicate find values or overlapping ranges in a rules table.

First and last matching

First or last match criteria are implemented depending on what you specified. The first match option looks for the first match in the relevant rules table. This method avoids duplicates in the find column of the rules table, or overlapping ranges in the discretize clean type. You can order the rules by specifying an order column. If you do not specify an order column, a default order is assumed. For example, in the find-and-replace clean type, the default order is the minimum replace value when an exact match is specified. If an exact match is not specified, the default order is the minimum find-and-replace value pair for a substring match. This behavior is also true for the last match where the maximum replace value is used.

Error on multiple matches

For the find-and-replace and discretize into ranges clean types, if an input value matches multiple rules, you can write the row to the error table. This method also implies that records that do not match any rule are also put into the error table.

Differentiator columns: You can optionally specify a differentiator column in the input table and the rules table. The differentiator column matches to satisfy the overall match of an input in the source table and a rule in the rules table.

Automatic summary tables: To support first and last match with high performance, you can use summary tables that are automatically created by the transformer. You can also provide your own names for the tables. This function is off by default.

To check that the summary table was used by DB2 for enhancing query performance, look at the SQL query that the Clean Data transformer generates in IWH.LOGTABLE. Use a tool such as Visual Explain to view the access plan. When the Find and Replace clean type is used with the replace substring option, you might need to edit your SQL query, because the REPLACE SQL function is not completely written to the log.

If a table exists with the summary table name that the transformer is trying to create, the summary table is not created. The error will be written to the

IWH.LOGTABLE, but because a summary table is not essential for the transformer to run successfully, the transformer does not stop running.

Error processing: Records that do not match rules are written to an error table. The records are written to the table unchanged and with a run ID.

Restrictions

The following restrictions apply to the Clean Data transformer:

- The source table and target table must be in the same database.
- The source table must be a single warehouse table.
- The target table is the default target table, or an existing table in the same target database.
- You can make changes to the step only when the step is in development mode.
- You cannot map a source column that allows null values to a target column that does not allow null values.
- The Null Replace Value field is not available for the Encode invalid values clean type. To replace nulls with the valid value that you specify, you should not have a null value in the rules table for Encode invalid.
- The Use minimum replacement value option for the Numeric Clip clean type is not available for z/OS.
- Some of the matching options will not work for either z/OS or iSeries™ because of the database server restrictions on the respective platforms.
- You must manually drop the summary tables and target error table from the target warehouse database when it is no longer required. This might be required when you are demoting the step to development.
- For the Find and Replace parameter: If your Clean Data transformer rules table contains different data types for the Find and Replace columns, you must change the Target column data type in the Target Table properties page and the Column Mapping page before you promote the transformer to test mode.
- For the Discretize parameter: If your Clean Data transformer rules table contains different data types for the Bounds and Replace columns, you must change the Target column data type in the Target Table properties page and the Column Mapping page before you promote the transformer to test mode.
- On an iSeries operating system, the Clean Data transformer does not perform error processing. You can generate the All Matches match type only on an iSeries operating system.

Related concepts:

- “Data Warehouse Center Version 7.2 Clean Data transformer” on page 249

Related tasks:

- “Specifying a numeric tolerance for your Clean Data transformer: Data Warehouse Center help”
- “Defining rules for a Clean Data transformer: Data Warehouse Center help”
- “Customizing the Clean Data transformer performance: Data Warehouse Center help”
- “Defining parameters for a Clean Data transformer: Data Warehouse Center help”

Rules tables for Clean Data transformers

The Clean Data transformer does not copy values that are not listed in the find column to the target table. Instead it copies the original source value to an error table if the user has selected the error processing option.

The following table describes the columns that might be included in the rules table for each clean type:

Table 25. Possible columns in the rules table for each clean type

Clean type	Minimum number of columns in rules table	Purpose of columns
Find and replace	2 - 4	<p>The first column contains find values.</p> <p>The second column contains replace values.</p> <p>The third column, if specified, contains rule-ordering information and is not required.</p> <p>The fourth column, if specified, contains a differentiator value which should match a differentiator value column in the source table.</p>
Discretize	3 - 5	<p>The first column contains the low end of the find value range.</p> <p>The second column contains the high end of the find value range.</p> <p>The third column contains the replacement value.</p> <p>The fourth column, if specified, contains rule-ordering information and is not required.</p> <p>The fifth column, if specified, contains a differentiator value which should match a differentiator value column in the source table.</p>
Numeric TM Clip	4 - 6	<p>The first column contains the low end of the find value range.</p> <p>The second column contains the high end of the find value range.</p> <p>The third column contains the low end of the replacement value.</p> <p>The fourth column contains the high end of the replacement value.</p> <p>The fifth column, if specified, contains rule-ordering information and is not required.</p> <p>The sixth column, if specified, contains a differentiator value which should match a differentiator value column in the source table.</p> <p>Tip: When you use the Numeric Clip clean type, you can run a find and replace procedure that searches outside of value ranges.</p>

Table 25. Possible columns in the rules table for each clean type (continued)

Clean type	Minimum number of columns in rules table	Purpose of columns
Carry over	None allowed	Rule table not used.
Convert case	None allowed	Rule table not used.
Encode invalid values	1	The first column is the column that contains the valid values.

You can reorder the output columns using the Properties notebook for the step. You can change column names on the Column Mapping page of the Properties notebook for the step.

Related concepts:

- “Clean Data transformer” on page 143

Related tasks:

- “Defining statistical transformers in the Data Warehouse Center” on page 163

Related reference:

- “Warehouse transformers” on page 96

Key columns

Use the Generate Key Table transformer to add a unique key to a warehouse table.

You can change values in the key column in two ways:

- You can change values based on a column of data type INT in the target table or another table in the warehouse target.
- You can replace all the key values in the key column of the target table.

The Generate Key Table transformer uses a warehouse target table as a source. The transformer writes to a table on the warehouse target. Before you define this step, link the warehouse target to the step in the Process Model window, with the arrow pointing towards the step. You can make changes to the step only when the step is in development mode.

Column mapping is not available for this step.

Period table

Use the Generate Period Table transformer to create a period table that contains columns of date information that you can use when evaluating other data, such as determining sales volume within a certain period of time.

You can use a period table in the following ways:

- Analyze trends by creating one or more output columns that are based on the generated date/time value for each row.
- Join a period table to other data tables. You can use the resulting table for simpler SQL SELECT statements that are based on user requirements.
- As a starting point for creating a more complex data table after the transformer adds other columns to the generated period table.

Transforming data

The Generate Period Table transformer works only on target tables. To use the transformer successfully, you must connect the transformer to a target.

You can make changes to the step definition only when the step is in development mode.

Related concepts:

- “Invert Data transformer” on page 150
- “Pivot data transformer” on page 151
- “Error logging for warehouse programs and transformers” on page 231
- “Clean Data transformer” on page 143

Related tasks:

- “Steps and tasks: Data Warehouse Center help”

Related reference:

- “Warehouse transformers” on page 96

Invert Data transformer

Use the Invert Data transformer to invert the order of the rows and columns in a table. When you use the Invert Data transformer, the rows in the source table become columns in the output table, and the columns in the input table become rows in the output table. The order of data among the columns, from top to bottom, is maintained and placed in rows, from left to right.

For example, consider the input table as a matrix. This transformer swaps the data in the table around a diagonal line that extends from the upper left of the table to the lower right. Then the transformer writes the transformed data to the target table.

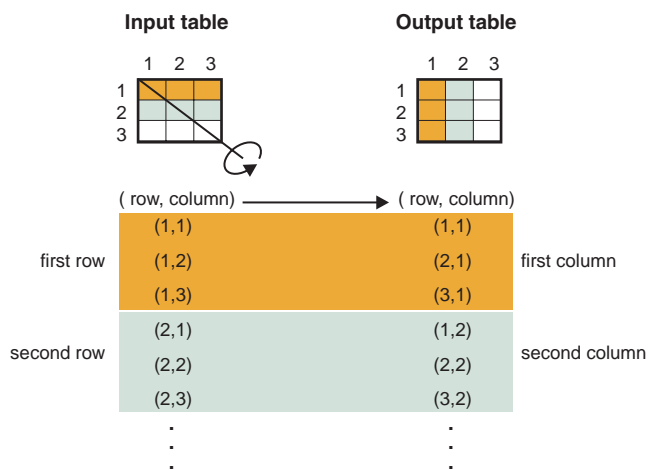


Figure 12. An example of the relationship between the input table and the output table from the Invert Data transformer

You can specify an additional column that contains ordinal data that starts at the number 1. This column helps you identify the rows after the transformer inverts the table.

You can also specify a column in the source table to be used as column names in the output table. This column is called the pivot column.

Columnar data in each pivot group must be either the same data type or data types that are related to each other through automatic promotion.

Before you define an Invert data step, you must connect a source table from the warehouse database to the step. You can also specify a target table that the step will write to, or you can designate that the step creates the target table. The desired output columns must be created manually in a step-generated target table.

The Invert Data transformer drops the existing database table and recreates it during each run. Each time that you run a step using this transformer, the existing data is replaced, but the table space and table index names are preserved.

A step that uses the Invert Data transformer must be promoted to production mode before you can see the actual data that is produced.

This step does not use the Column Mapping page.

Related tasks:

- “Transforming target tables” on page 143
- “Steps and tasks: Data Warehouse Center help”

Related reference:

- “Warehouse transformers” on page 96

Pivot data transformer

Use the Pivot data transformer to group related data from selected columns in the source table, which are called pivot columns, into a single column, called a pivot group column, in the target table. You can create more than one pivot group column.

You can select multiple columns from the source table to carry over to the output table. The data in these columns is not changed by the Pivot data transformer.

You can specify an additional column that contains ordinal data that starts at the number 1. This column helps you identify the rows after the transformer inverts the table.

Columnar data in each pivot group must have the same data type or data types that are related to each other through automatic promotion.

Before you begin this task, connect a warehouse source table to the step in the Process Model window. The Pivot Data transformer uses an existing target table in the same database or creates a target table in the same database that contains the warehouse source. You can change the step definition only when the step is in development mode.

Related concepts:

- “Invert Data transformer” on page 150
- “Error logging for warehouse programs and transformers” on page 231
- “Clean Data transformer” on page 143

Transforming data

Related tasks:

- “Steps and tasks: Data Warehouse Center help”

Related reference:

- “Warehouse transformers” on page 96

FormatDate transformer

The FormatDate transformer provides several standard date formats that you can specify for the input and output columns. If a date in the input column does not match the specified format, the transformer writes a null value to the output table.

If the format that you want to use is not displayed in the **Format** list, you can type a format in the **Format string** field of the transformer window. For example, type **MMM D, YY** if the dates in your input column have a structure like Mar 2, 96 or Jul 15, 83.

The data type of the **Output column** field is VARCHAR(255). You cannot change the data type by selecting Date, Time, or Date/Time from the **Category** list on the Function Parameters - FormatDate page.

The date in the Modified column is displayed in the format MM/DD/YYYY and includes the time, such as 06/17/2003 2:47:15 PM. You cannot change the format for this date and time field.

Changing the format of a date field

Use the FormatDate transformer to change the format of a date field in your source table that your step is copying to the default target table. You can run this transformer with any other transformer or warehouse program.

Procedure:

To use the FormatDate transformer:

1. Create a step that generates a default target table in a warehouse target where the transformers are installed.
2. Link to the source table.
3. Click **Build SQL** on the SQL Statement page of the Properties notebook for the step.
4. Click the **Columns** tab.
5. Click **Add**.
The Expression Builder opens.
6. In the Expression Builder, double-click **FormatDate** in the list of function names.
The Function Arguments - FormatDate window opens.
7. Click the column in the **Available columns** list that contains date or time information that you want to reformat, then click > to place the selected field in the **Input column**.
8. Specify the input format in the **Input Format** area of the window.
9. Specify the output format in the **Output format** area of the window.
10. Click **OK**.

The Function Arguments - FormatDate window closes, and the expression is displayed in the Expression Builder window.

Related concepts:

- “FormatDate transformer” on page 152

Related tasks:

- “Specifying the input format for a date field” on page 153
- “Specifying the output format for a date field” on page 153

Specifying the input format for a date field

You can specify the input format for a date field in the properties notebook for the FormatDate transformer.

Procedure:

1. From the Function Arguments — FormatDate window Expression Builder, select a category for the input column data from the **Category** list.
2. Select a date, time, or timestamp format from the **Format** list. The **Example** list shows an example of the format that you select. The **Format string** field confirms your selection. You can also specify a format by typing it in the **Format string** field.

Specifying the output format for a date field

You can specify the output format for a date field when you’re using the FormatDate transformer.

Procedure:

To specify the output format for a date field:

1. From the Function Arguments — FormatDate window Expression Builder, select a category for the output column data from the **Category** list.
2. Select a date, time, or timestamp format from the **Format** list. The **Example** list shows an example of the format that you select. The **Format string** field confirms your selection. You can also specify a format by typing it in the **Format string** field.

Cleaning name and address data

These sections explain how to clean name and address data in the Data Warehouse Center using the Trillium Software System. The Data Warehouse Center already provides integration with tools from Vality and Evolutionary Technologies, Inc.

Cleaning name and address data with the Trillium Software System

Use the Data Warehouse Center and the Trillium Software System to cleanse name and address data. The Trillium Software System is a name and address cleansing product that reformats, standardizes, and verifies name and address data. You can use the Trillium Software System in the Data Warehouse Center by starting the

Transforming data

Trillium Batch System programs from a user-defined program. The user-defined program is added to the Warehouse tree when you import the metadata from the Trillium Batch System script or JCL.

Prerequisites:

- You must install the Trillium Software System on the warehouse agent site or on a remote host.
- On UNIX and Windows systems, you must add the path to the bin directory for the Trillium Software System to the PATH environment variable to enable the agent to run the Trillium Batch System programs. On UNIX, you must add the PATH variable to the IWH.environment file before you start the vwdaemon process.

The following table shows the software that is required for cleaning name and address data in the Data Warehouse Center.

Table 26. Required software

Operating system	Required software
UNIX	Trillium Software System Version 4.0
	DB2 Universal Database Warehouse Manager Version 7.2 or later warehouse agent
Windows	Trillium Software System Version 4.0
	DB2 Universal Database Warehouse Manager Version 7.2 or later warehouse agent
	For remote access, the host system must have ftpd and rexecd daemons.
z/OS	Trillium Software System Version 4.0 installed on the remote z/OS host
	DB2 Universal Database Warehouse Manager Version 7.2 or later warehouse agent installed on UNIX, or Windows
	TCP/IP 3.2 or later
	The z/OS operating system is supported as a remote host only

Restrictions:

You can specify overlapping fields in the input and output DDL files with the Trillium DDL and the import metadata operation in the Data Warehouse Center. However, the corresponding warehouse source and warehouse target files cannot be used in the Data Warehouse Center with an SQL step, or with the Sample Contents. Because the import metadata operation ignores overlapping fields that span the whole record, you can still specify these fields, but they will not be used as columns in the resulting source and target files.

If an error file is specified, the name of the script cannot contain any blank spaces.

Procedure:

To clean name and address data:

1. Import the Trillium metadata that you want to use.
2. Map the metadata to Data Warehouse Center data types.

3. Define a Trillium Batch System step.
4. Add the Trillium Batch System step to a process.

Related concepts:

- “Trillium Batch System JCL files” on page 157
- “Trillium Software System components” on page 155
- “Trillium metadata” on page 156
- “Trillium Batch System user-defined program” on page 158
- “Error handling for Trillium Batch System programs” on page 160

Related tasks:

- “Importing Trillium metadata” on page 156

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Parameters for the Trillium Batch System script or JCL” on page 159
- “Error return codes for name and address cleansing” on page 161

Trillium Software System components

The Trillium Software System consists of four main components: converter, parser, geocoder, and matcher. Use the components as a set of functions to perform name and address cleansing operations. You can run the components from the Trillium Batch System, which is a user-defined program.

Converter

Use the converter to standardize and convert the source data into the specified output format.

Parser Use the parser to interpret name and address source data and create metadata about the source data.

Geocoder

Use the geocoder to compare the source data with postal service data to supply any missing information such as courier or ZIP+4 codes. The geocoder also performs matching operations with United States Census data.

Matcher

Use the matcher to compare similar names and addresses to identify duplicate records. You can perform reference matching by using the matcher to compare one record to a group of records.

Related concepts:

- “Trillium Batch System JCL files” on page 157
- “Trillium metadata” on page 156
- “Trillium Batch System user-defined program” on page 158
- “Error handling for Trillium Batch System programs” on page 160

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153
- “Importing Trillium metadata” on page 156

Transforming data

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Parameters for the Trillium Batch System script or JCL” on page 159

Trillium metadata

In the Data Warehouse Center, you can import Trillium Batch System metadata and create a user-defined program step. This step calls a Trillium Batch System script on the local warehouse agent site, or on a remote warehouse agent site. In the Data Warehouse Center, the Trillium Batch System script is a step with a source and target file. The source file is the input data file that is used for the first Trillium Batch System command. The target file is the output data file that is created by the last Trillium command in the script. The step can then be copied to another process to be used with other steps.

The following figures show the relationship between the Trillium Batch System input and output data files and the source and target files in the Data Warehouse Center.

```
REM Running the converter
pfcondrv -parmfile c:\tril40\us_proj\parms\pfcondrv.par
REM Running the parser
pfprsdrv -parmfile c:\tril40\us_proj\parms\pfprsdrv.par
REM Running the Matcher
cfmatdrv -parmfile c:\tril40\us_proj\parms\pfmatdrv.par
```

Figure 13. Sample Trillium script file

```
INP_FNAME01 c:\tril40\us_proj\data\convinp
INP_DDL01 c:\tril40\us_proj\dict\input.ddl
```

Figure 14. Contents of the pfcondrv.par file

```
OUT_DDNAME c:\tril40\us_proj\data\maout
DDL_OUT_FNAME c:\tril40\us_proj\dict\parseout.ddl
```

Figure 15. Contents of the pfmatdrv.par file

```
c:\Tril40\us_proj\data\convinp (source file) ->
Trillium Batch System Step -> c:\tril40\us_proj\data\maout (target file)
```

Figure 16. The Trillium Batch System step definition

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237

Importing Trillium metadata

In addition to importing other kinds of metadata, you can import Trillium metadata and use it for cleaning name and address data.

Restrictions:

When you create the Trillium Batch System user-defined program step using the Import Metadata notebook for the Trillium Batch System, you must select **Remote**

host as your connection for the zSeries agent, even when the JCL is on the same system as the agent. All parameters for the remote host connection must be entered.

After you create the Trillium Batch System user-defined program step, use the Properties notebook of the Trillium Batch System step to change the agent site to the zSeries agent site that you want to use.

If the name of the JCL or the output error file contains any blanks or parentheses, you must enclose them in quotation marks when you enter them into the Script, JCL, or Output error file fields.

Procedure:

To import Trillium metadata:

1. Create a Trillium Batch System script or JCL.
2. To open the Import Metadata Trillium window, expand the Data Warehouse Center navigation tree, right-click **Warehouse**, and select **Import Metadata** → **Trillium**.
3. Specify the information for your import.

Related concepts:

- “Trillium Batch System JCL files” on page 157
- “Trillium Software System components” on page 155
- “Trillium metadata” on page 156
- “Trillium Batch System user-defined program” on page 158
- “Error handling for Trillium Batch System programs” on page 160

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Parameters for the Trillium Batch System script or JCL” on page 159

Trillium Batch System JCL files

The following requirements must be met if you are writing a Trillium Batch System JCL file.

- The job name must be the user ID plus one character.
- The job must be routed to the held output class.
- Each job step that runs a Trillium Batch System program must include a SYSTERM DD statement that defines a permanent data set. The data set contains the errors from the Trillium Batch System programs. This data set will be automatically deleted before the JCL is submitted.

The output error file must be specified when the script or JCL runs on a remote host; otherwise, the error messages will not be captured and returned to the Data Warehouse Center. On UNIX[®] or Windows[®], the simplest way to capture the error messages is to write another script that calls the Trillium Batch System script and pipes the standard error to an output file.

Writing Trillium Batch System script file on UNIX and Windows

If the Trillium Batch System script or parameter files contain relative paths of input files, the user must put a `cd` statement at the beginning of the script file to the directory of the script file.

Related concepts:

- “Trillium Software System components” on page 155
- “Trillium metadata” on page 156
- “Trillium Batch System user-defined program” on page 158
- “Error handling for Trillium Batch System programs” on page 160

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153
- “Importing Trillium metadata” on page 156

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Parameters for the Trillium Batch System script or JCL” on page 159

Example: Job step that includes a SYSTERM DD statement

This is an example of a job step that includes a SYSTERM DD statement.

```
//SYSTERM DD UNIT=&UNIT,  
//          DISP=(MOD,CATLG,KEEP),  
//          SPACE=(400,(20,20),,ROUND),  
//          DSN=&PROJPREF.&TRILVER.&PROJECT.STDERR;
```

Trillium Batch System user-defined program

The Trillium Batch System user-defined program is included with DB2® Data Warehouse Center Version 7.2 or later for Windows® NT, Windows 2000, Windows XP and UNIX®. The Trillium Batch System step that is created when you import Trillium metadata will run the Trillium Batch System user-defined program. The user-defined program will call the Trillium Batch System script or JCL.

Related concepts:

- “Trillium Batch System JCL files” on page 157
- “Trillium Software System components” on page 155
- “Trillium metadata” on page 156
- “Error handling for Trillium Batch System programs” on page 160

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153
- “Importing Trillium metadata” on page 156

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Parameters for the Trillium Batch System script or JCL” on page 159

Parameters for the Trillium Batch System script or JCL

The following table contains the parameters for the Trillium Batch System script or JCL. The data type for all of the parameters in this table is CHARACTER.

Table 27. Parameters for the Trillium Batch System script or JCL

Parameter	Values
Remote host	<ul style="list-style-type: none"> localhost is the default value. Use this value if the Trillium Batch System is installed on the warehouse agent site. The name of the remote host if the Trillium Batch System is installed on a remote operating system.
Script or JCL	The name of the script or JCL.
Remote operating system	<p>The name of the operating system on the remote host. This parameter is ignored if the value of the Remote host parameter is localhost. The valid values are:</p> <ul style="list-style-type: none"> MVS for the z/OS operating system UNIX for AIX, Solaris Operating Environment, HP-UX, and NUMA/Q operating systems WIN for the Windows NT, Windows 2000, Windows XP operating systems
Remote user ID	The user ID with the authority to issue the remote command. This parameter is ignored if the value of Remote host is localhost.
Password option	<p>The method to obtain the password. The valid values are:</p> <p>ENTERPASSWORD Use this value if the password is passed in the next parameter.</p> <p>PASSWORDNOTREQUIRED Use this value if no password is needed.</p> <p>GETPASSWORD Use this value if a program name is passed in the next parameter.</p> <p>Requirements:</p> <ul style="list-style-type: none"> The program must reside on the agent site, write the password to an output file in the first line, and return 0 if it runs successfully. The value of the Password parameter must be the name of the password program. The value of the Program parameters parameter must be a string enclosed in quotation marks. The first parameter in the string must be the name of the output file where the password will be written.
Password	The valid value is the password or password program name. The password program must be local to the warehouse agent site.
Program parameters	The parameters for the password program.
Output Error File	The name of the output error file.

Related concepts:

- “Trillium Batch System JCL files” on page 157
- “Trillium Software System components” on page 155
- “Trillium metadata” on page 156

Transforming data

- “Trillium Batch System user-defined program” on page 158
- “Error handling for Trillium Batch System programs” on page 160

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153
- “Importing Trillium metadata” on page 156

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237

Error handling for Trillium Batch System programs

The Trillium Batch System programs write error messages to the standard error (stderr) file on the Windows[®] NT, Windows 2000, Windows XP and UNIX[®] operating systems, and to the SYSTERM data set on the z/OS[™] operating system.

To capture the errors from the Trillium Batch System programs on Windows NT or UNIX operating systems, the standard error must be redirected to an output error file.

To capture the errors from the Trillium Batch System programs on the z/OS operating system, the JCL must include a SYSTERM DD statement.

If you specify the output error file name in the Import Metadata window, you must redirect, or store the standard error output to the error file. The Data Warehouse Center will read the file and report all of the lines that contain the string ERROR as error messages. All of the Trillium Batch System program error messages contain the string ERROR.

If the output error file is not specified in a script or in JCL running on the warehouse agent site, the Data Warehouse Center will automatically create a file name and redirect the standard error output to that file. If an error is encountered, the error file will not be deleted. The error file is stored in the directory specified by the environment variable VWS_LOGGING. The file name is *tbsudp-date-time.err*, where *date* is the system date when the file is created, and *time* is the system time when the file is created. The following file name shows the format of the output error file name:

```
tbsudp-021501-155606.err
```

Related concepts:

- “Trillium Batch System JCL files” on page 157
- “Trillium Software System components” on page 155
- “Trillium metadata” on page 156
- “Trillium Batch System user-defined program” on page 158

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153
- “Importing Trillium metadata” on page 156

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237

- “Parameters for the Trillium Batch System script or JCL” on page 159

Error return codes for name and address cleansing

The following table lists the return codes for Trillium Batch System programs.

Table 28. Return codes for Trillium Batch System programs

Error number	Description
0	Success
4	Warning. Either the password file could not be erased or an internal error occurred while the Trillium Batch System user-defined program was accessing a temporary file. Check the status of the password file or all of the temporary files created under the directory that is specified by the environment variable VWS_LOGGING.
8	The amount or values of the parameters are not correct. Read the log file or the documentation for the correct syntax.
12	A problem occurred while the Trillium Batch System user-defined program was connecting to the remote host through FTP. Check the FTP connection or host name, user ID, and password.
16	The Trillium Batch System user-defined program cannot create the log or an internal file. Check to see if the user has the correct authorization. Check to see if the disk is full.
20	Either the z/OS JCL cannot be executed, or an error occurred while the Trillium Batch System user-defined program was deleting or getting a file from z/OS through FTP. Check the JESLogFile to identify the reason.
48	Environment variable VWS_LOGGING cannot be found or the log file cannot be created. Check the log file for more information.
56	Either the Windows NT or UNIX script cannot be run, or an error occurred while the Trillium Batch System user-defined program was connecting to the remote host. Check the connection or host name, user ID, and password.
500	The script or JCL file returns an error, or it does not return an error but the error file contains data. Check the log file for more information. On z/OS, also check the JESLogFile.

Log files

The Data Warehouse Center stores all diagnostic information in a log file when the Trillium Batch System user-defined program runs. The name of the log file is *tbsudp-date-time.log*, where *date* is the system date when the file is created, and *time* is the system time when the file is created. The log file is created in the directory that is specified by the environment variable VWS_LOGGING on the agent site. The log file is deleted if the Trillium Batch System user-defined program runs successfully.

Related concepts:

- “Trillium Batch System user-defined program” on page 158
- “Error handling for Trillium Batch System programs” on page 160

Transforming data

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153

Chapter 13. Calculating statistics

The chapter describes how to use transformers to perform statistical functions.

Defining statistical transformers in the Data Warehouse Center

The following statistical transformers are available to you in the Data Warehouse Center:

- ANOVA
- Calculate Statistics
- Calculate Subtotals
- Chi-square
- Correlation
- Moving Average
- Regression

Procedure:

To perform a statistical transformation on your data, define the statistical transformer that you want to use.

Related concepts:

- “Chi-square transformer” on page 166

Related tasks:

- “Steps and tasks: Data Warehouse Center help”

ANOVA transformer

Use the Analysis of Variance (ANOVA) transformer to produce statistical calculations in two tables based on a small number of parameters. There are three types of ANOVA: one-way ANOVA, two-way ANOVA, and three-way ANOVA.

The ANOVA transformer obtains two independent estimates of variance. The first estimate is based on variability between groups. The second estimate is based on variability within groups. After the ANOVA transformer computes these estimates, it calculates their ratio. A family of distributions, the Fisher-F distributions, describes the significance of this ratio.

This transformer also calculates a p-value. The p-value is the probability that the means of the two groups are equal. A small p-value leads to the conclusion that the means are different. For example, a p-value of 0.02 means that there is a 2% chance that the sample means are equal. Likewise, a large p-value leads to the conclusion that the means of the two groups are not different.

You can use this step only with tables that exist in the same database. Use a warehouse source or target table as a source for the ANOVA transformer and up to two warehouse target tables as targets for the ANOVA statistical calculations. If you do not want to select a target table for the ANOVA transformation, you can

Calculating statistics

specify that the ANOVA transformer creates tables on the target database. The Parameters page will not be available for this step subtype until you link the step to a source in the Process Model window.

Each time you run a step using this transformer, the existing data is replaced. The ANOVA transformer drops the existing database table and recreates it during each run.

You can make changes to the step only when the step is in development mode.

Related tasks:

- “Creating default target tables for an ANOVA transformer: Data Warehouse Center help”

Calculate Statistics transformer

Use the Calculate Statistics transformer to calculate the following descriptive statistics on any number of data columns from a single table:

- Count
- Sum
- Average
- Variance
- Standard deviation
- Standard error
- Minimum
- Maximum
- Range
- Coefficient of variation

To use the Calculate Statistics transformer, connect the step to a warehouse source and a warehouse target that exist in the same database. Or, you can connect the step to a warehouse source and specify that the step create a target table for you in the same database.

You can make changes to the step only when the step is in development mode.

Related tasks:

- “Calculating common descriptive statistics: Data Warehouse Center help”

Calculate Subtotals transformer

Use the Calculate subtotals transformer to calculate the running subtotal for a set of numeric values grouped by a period of time, either weekly, semimonthly, monthly, quarterly, or annually. For example, for accounting purposes, it is often necessary to produce subtotals of numeric values for basic periods of time. This is most frequently encountered in payroll calculations where companies are required to produce month-to-date and year-to-date subtotals for various types of payroll data.

The Calculate Subtotals transformer uses a warehouse target table as a source. The table that you use as a source must contain a primary key. If you use a target table that was generated by the Data Warehouse Center, you must assign a primary key

to the table before you use it as a source. The transformer writes to a table on the same database. Before you define this step, link the warehouse target to the step in the Process Model window, with the arrow pointing towards the step. The Parameters page will not be available for this step subtype until you link it to a source in the Process Model window. You can make changes to the step only when the step is in development mode.

The following restrictions apply to a single instance of a step. For example, you have step 1 and step 2. In step 1, you can use column B as a source column. In step 2, you can use column B as a target column.

- Because the calculate subtotal transformer can write to its source column, you can map an input column to itself. For example, you can map source column A to itself:

Table 29. Source column mapped to itself

Source column	Target column
A	A

You can map a column to itself only if the column is not used as an input column in another transformer definition row. For example, you cannot map column A to itself if the following is true:

Table 30. Source column used as an input column

Source column	Target column
A_week	
A_month	

In this example, column A is used as an input column in two separate transformer definitions.

- You cannot use a source column that is mapped to a target column as a target column in another mapping in the same step definition. For example, you have the columns B, C and D. Your first row is specified as follows:

Table 31. Source column mapped to a target column

Source column	Target column
B	C

Because column B is mapped as a source column, the following mapping is not allowed:

Table 32. Mapping that is not allowed

Source column	Target column
D	B

- After a column is mapped as a target, you cannot use the column as either an input column or a target output column in any other mappings in this step definition. For example, you have the following rows:

Table 33. Example of column mapping

Source column	Target column
A	A
B	C

Because you specified Column A and Column C as output columns, you cannot use them as either input columns or output columns in any additional rows.

Calculating statistics

Related tasks:

- “Calculating the running subtotal for numeric values grouped by period: Data Warehouse Center help”
- “Reference for Calculate subtotals: Data Warehouse Center help”

Chi-square transformer

Use the Chi-Square transformer to perform the chi-square test and the chi-square goodness-of-fit test on columns of numerical data. These tests are nonparametric tests.

You can use the statistical results of these tests to make the following determinations:

- Whether the values of one variable are related to the values of another variable
- Whether the values of one variable are independent of the values of another variable
- Whether the distribution of variable values meets your expectations

Use these tests with small sample sizes or when the variables that you are considering might not be normally distributed. Both the chi-square test and the chi-square goodness-of-fit test make the best use of data that cannot be precisely measured.

The Chi-square transformer optionally produces an additional output table called the Expected value output table. You can select a table to be used as the Expected value output table, or you can specify that the table will not be produced.

When you set up this process in the Process Model window, link the Chi-Square step to a warehouse target table. If you want the step to produce the Expected value output table, link the step to a second warehouse target table in the same database.

You can change the step definition only when the step is in development mode.

Related tasks:

- “Performing chi-square tests: Data Warehouse Center help”
- “Creating default target tables for a Chi-Square transformer: Data Warehouse Center help”
- “Steps and tasks: Data Warehouse Center help”
- “Defining statistics calculations: Data Warehouse Center help”

Correlation analysis

Use the Correlation transformer to determine the extent to which changes in the value of an attribute (such as length of employment) are associated with changes in another attribute (such as salary). The data for a correlation analysis consists of two input columns. Each column contains values for one of the attributes of interest. The Correlation transformer can calculate various measures of association between the two input columns. You can select more than one statistic to calculate for a given pair of input columns.

The data in the input columns also can be treated as a sample obtained from a larger population, and the Correlation transformer can be used to test whether the attributes are correlated in the population. In this context, the *null hypothesis* asserts that the two attributes are not correlated, and the *alternative hypothesis* asserts that the attributes are correlated.

The Correlation transformer calculates any of the following correlation-related statistics on one or more pairs of columns:

Correlation coefficient r

The correlation coefficient r is a measure of the linear relationship between two attributes or columns of data. The correlation coefficient is also known as the Pearson product-moment correlation coefficient. The value of r can range from -1 to +1 and is independent of the units of measurement. A value of r near 0 indicates little correlation between attributes; a value near +1 or -1 indicates a high level of correlation.

When two attributes have a positive correlation coefficient, an increase in the value of one attribute indicates a likely increase in the value of the second attribute. A correlation coefficient of less than 0 indicates a negative correlation. That is, when one attribute shows an increase in value, the other attribute tends to show a decrease.

Consider two variables x and y :

- If $r = 1$, then x and y are perfectly positively correlated. The possible values of x and y all lie on a straight line with a positive slope in the (x,y) plane.
- If $r = 0$, then x and y are not correlated. They do not have an apparent linear relationship. However, this does not mean that x and y are statistically independent.
- If $r = -1$, then x and y are perfectly negatively correlated. The possible values of x and y all lie on a straight line with a negative slope in the (x,y) plane.

Covariance

Covariance is a measure of the linear relationship between two attributes or columns of data. The value of the covariance can range from $-\infty$ to $+\infty$. However, if the value of the covariance is too small or too large to be represented by a number, the value is represented by NULL.

Unlike the correlation coefficient, the covariance is dependent on the units of measurement. For example, measuring values of two attributes in inches rather than feet increases the covariance by a factor of 144.

T-value

T-value is the observed value of the T-statistic that is used to test the hypothesis that two attributes are correlated. The T-value can range between $-\infty$ and $+\infty$. A T-value near 0 is evidence for the null hypothesis that there is no correlation between the attributes. A T-value far from 0 (either positive or negative) is evidence for the alternative hypothesis that there is correlation between the attributes.

The definition of T-statistic is:

$$T = r * \text{SQRT}((n-2) / (1 - r*r))$$

where r is the correlation coefficient, n is the number of input value pairs, and SQRT is the square root function.

Calculating statistics

If the correlation coefficient r is either -1 or +1, the T-value is represented by NULL. If the T-value is too small or too large to be represented by a number, the value is represented by NULL.

P-value

P-value is the probability, when the null hypothesis is true, that the absolute value of the T-statistic would equal or exceed the observed value (T-value). A small P-value is evidence that the null hypothesis is false and the attributes are, in fact, correlated.

Your source table and target table must exist in the warehouse database. This transformer can create a target table in the same warehouse database that contains the source, if you want it to. You can change the step only when the step is in development mode.

Related tasks:

- “Defining statistical transformers in the Data Warehouse Center” on page 163
- “Determining the linear relationship between variables in two columns: Data Warehouse Center help”
- “Finding relationships and correlation between two variables: Data Warehouse Center help”
- “Defining correlation statistics: Data Warehouse Center help”

Moving averages

Simple and exponentially smoothed moving averages can often predict the future course of a time-related series of values. Moving averages are widely used in time-series analysis in business and financial forecasting. Rolling sums have other widely used financial uses.

You can use the Moving Average transformer to calculate the following values:

- A simple moving average
- An exponential moving average
- A rolling sum for N periods of data, where N is specified by the user

An exponential moving average is also known as an exponentially smoothed moving average.

Moving averages redistribute events that occur briefly over a wider period of time. This redistribution serves to remove noise, random occurrences, large peaks, and valleys from time-series data. You can apply the moving average method to a time-series data set to:

- Remove the effects of seasonal variations.
- Extract the data trend.
- Enhance the long-term cycles.
- Smooth a data set before performing higher-level analysis.

The Moving Average transformer uses a warehouse target table as a source. The table that you use as a source must contain a primary key. If you use a target table that was generated by the Data Warehouse Center, you must assign a primary key to the table before you use it as a source. The transformer writes to a table on the warehouse target. Before you define this step, link the warehouse target to the step in the Process Model window, with the arrow pointing towards the step.

Related concepts:

- “ANOVA transformer” on page 163
- “Calculate Statistics transformer” on page 164
- “Calculate Subtotals transformer” on page 164
- “Chi-square transformer” on page 166
- “Regression transformer” on page 169
- “Error logging for warehouse programs and transformers” on page 231

Related tasks:

- “Defining statistical transformers in the Data Warehouse Center” on page 163

Related reference:

- “Warehouse transformers” on page 96
- “Statistical transformers” on page 96

Regression transformer

Use the Regression transformer to identify the relationships between a dependent variable and one or more independent variables, and to show how closely they are correlated. You can use this transformer to show the effect of a change in pricing on demand for a product, to show the effect of location on the response to advertising, or to show how closely two seemingly random sets of data are related.

This transformer performs a backward, full-model regression. This method starts with all independent variables in a model, but removes the least-important independent variables one at a time until only significant independent variables remain in the model.

The Regression transformer produces two additional output tables: the ANOVA summary table and the Equation variable table.

Before you begin this task, you must link this step in the Process Model window to a warehouse source table and three warehouse target tables. Or, you can link the step to a source and specify that the step create the target tables. The tables must exist in the same database. The Regression transformer writes the results from the Regression transformation to a table on one warehouse target, and creates the ANOVA summary table and the Equation variable table on the second and third targets. You can make changes to the step only when the step is in development mode.

Related concepts:

- “ANOVA transformer” on page 163
- “Calculate Statistics transformer” on page 164
- “Calculate Subtotals transformer” on page 164
- “Chi-square transformer” on page 166
- “Moving averages” on page 168

Related tasks:

- “Defining statistical transformers in the Data Warehouse Center” on page 163

Calculating statistics

Chapter 14. Exporting and importing metadata in the Data Warehouse Center

This chapter describes how to use the Data Warehouse Center metadata export and import capabilities to export and import object definitions. Starting in the Data Warehouse Center Version 8.2, the .INP files that are generated when you run the EXPORT utility from the GUI will be in Unicode format.

Metadata export capabilities

You can use the Data Warehouse Center export capabilities to export subjects, processes, sources, targets and user-defined program definitions. When an object is exported, all of the dependent and subordinate objects will be exported to the tag language file or XML file by default. The Data Warehouse Center export capabilities are available for the following operating systems:

- AIX
- Solaris Operating Environment
- Windows

You can export the following types of metadata:

- Tag language
- Common Warehouse Metamodel metadata
- OLAP Integration Server metadata (Windows only)

Exporting processes use a large amount of system resources. You might want to limit the use of other programs while you are exporting object definitions. When you do large export operations, you might want to increase the DB2 application heap number size of the warehouse control database to 8192.

Because the import and export formats are release-dependent, you cannot use exported files from a previous release to migrate from one release of the Data Warehouse Center to another.

Starting with the Data Warehouse Center, Version 8.2, tag language files generated by the EXPORT utility will be in Unicode format (UTF-8 encoding). In addition, the .INP files generated by the EXPORT utility will be in Unicode format. You must use a Unicode editor to view these generated files. The .INP file is generated when you run the EXPORT utility from the GUI. If you run the EXPORT utility from a command line, you must create the .INP file. The Data Warehouse Center supports both Unicode and non-Unicode versions of .INP files. If a byte order mark (BOM), which contains the character code U+FFFE, is found at the beginning of the file, then the Data Warehouse Center assumes that the .INP file is in Unicode format. Otherwise, the Data Warehouse Center will assume system default encoding when reading an .INP file.

By default, exporting includes the selected objects and all of the objects to which the selected objects refer. For example, if you select a process to export, the sources and targets used by the steps, dependent steps, and dependent processes are included. You can exclude the source definitions in an export (by specifying /R if

Exporting and importing metadata in the Data Warehouse Center

you are exporting from the command line). However, you must define the sources in the target system before you import the tag file to avoid an error.

You can limit the number of exported objects to decrease the size of the tag file. By default, the export operation includes steps that have a data dependency. For example, consider the following scenario: Process P1 contains step S1 that populates T1, and process P2 contains step S2 that contains a source as T1 so that the following dependency is established: S1 → T1 → S2 → T3. If you export only process P2, P1 is exported in the tag file too, because S2 is dependent on S1 for its data. The data dependency goes backwards. So, even if you export P2 only, P1 is included in the tag file too. Exporting P1 and P2 separately does not help, so the best approach is to export them together. If you are exporting from the command line, specify the /B option to exclude dependent steps from processes that are not selected.

In addition to data dependency, you also must consider cascading. Consider a step in process P5 with a shortcut to a step in process P6. If P5 is exported, P6 is exported too. In this case, exporting cascades downward to the next step through a shortcut. If you are exporting from the command line, the following options are available:

- Specify the /C option to exclude steps that are connected by shortcuts from the selected processes and processes that are connected by task flow. Use options /B and /C together to export only the specified processes, their sources, and targets into a tag file.
- Specify the /D option to exclude cascaded steps and processes, but include shortcuts to processes that are connected by task flow. You must define the target steps of shortcuts or process task flow in the target system before you import the tag file to avoid an error.

Related tasks:

- “Exporting metadata into a tag language file” in the *Data Warehouse Center Application Integration Guide*

Tag language and XML files

When you export metadata to a tag language file or XML file, the Data Warehouse Center finds the objects that you want to export and produces tag language or XML statements to represent the objects. It then places the statements into files that you can import into another Data Warehouse Center.

Several files can be created during a single export process. For example, when you export metadata definitions for BLOB data, multiple files are created. The first file created in the export process has an extension of .tag or .xml. If multiple files are created, the file name that is generated for each supplementary file has the same name as the tag language file with a numeric extension.

For example, if the tag language file name that you specified is e:\tag\steps.tag, the supplementary files are named e:\tag\steps.1, e:\tag\steps.2, and so on. Only the file extension is used to identify the supplementary files within the base tag language file, so you can move the files to another directory. However, you should not rename the files. You must always keep the files in the same directory, otherwise you will not be able to import the files successfully.

You can move tag language files from one operating system to another. However, you must transfer files with the .tag extension in ASCII mode. Similarly, you must transfer BLOB files, which have a numeric extension, in binary mode.

Tag language files and XML files might not be compatible between releases.

Importing metadata

You can import object definitions for use in your Data Warehouse Center system. For example, you might want to import sample data into a warehouse, or you might want to import metadata if you are creating a prototype for a new warehouse.

When you import metadata, all of the objects are assigned to the security group that is specified in the tag language file or XML file. If no security group is specified, all of the objects are assigned to the *default security group*.

Import processes use a large amount of system resources. You might want to limit the use of other programs while you are importing object definitions. When you do large import operations, you might want to increase the DB2 application heap number size of the warehouse control database to 8192.

You can import the following types of metadata:

- Tag language file
- Common Warehouse Metamodel metadata
- ERwin
- MQSeries
- Trillium

You can use the Data Warehouse Center import capabilities to import object definitions within the following operating systems:

- AIX
- Solaris Operating Environment
- Windows

If a process with unlinked shortcuts is exported and then imported into another control database as a .tag file, the unlinked shortcut data will cause error DWC3142: "<dirID> was not found in the Data Warehouse Center control database." This error displays when the unlinked shortcut dirIDs are not translated, and they refer back to the original control database.

If you move a tag language file or XML file from one system to another, you must move all the associated files with it, and they must reside in the same directory.

If you are using the import utility to establish a new Data Warehouse Center, you must initialize a new warehouse control database in the target system. After you complete this task, you can import as many tag language files or XML files as you want.

Starting in the Data Warehouse Center Version 8.2, the feedback files generated by the import utility will be in Unicode format. You must use Unicode editors to read and edit these feedback files.

Exporting and importing metadata in the Data Warehouse Center

Procedure:

To import metadata into the Data Warehouse Center:

1. Open the Import Metadata window for the type of metadata that you want to import.
2. Specify parameters for the import.

Related tasks:

- “Importing objects: Information Catalog Center help”
- “Importing an ERwin tag language file: Data Warehouse Center help”
- “Importing an MQ Series file: Data Warehouse Center help”

Related reference:

- “Metadata mappings between the Data Warehouse Center and CWM XML objects and properties” on page 238

Publishing warehouse metadata

Publishing metadata is the process of transferring metadata from the Data Warehouse Center to the Information Catalog Center.

If the warehouse server and warehouse agent are running on a remote system, you must catalog the warehouse control database with the same user ID, password, and name on the warehouse default agent system and the warehouse administrative client system. If the control database is not cataloged with the same name on both systems, the warehouse agent cannot communicate with the warehouse control database.

Prerequisites:

If you are using Data Warehouse Center publish to Information Catalog Center, you must update the Classpath and Path variables in the IWH.environment file. For example:

```
#CLASSPATH=/home/db2inst/sqllib/java/db2java.zip:/home/db2inst/sqllib/java/sqlj.zip:/home/db2inst/sqllib/java/runtime.zip:/home/db2inst/sqllib/java/Common.jar:/home/db2inst/sqllib/tools/db2_vw.jar:/home/db2inst/sqllib/tools/db2vwcom.jar:/home/db2inst/sqllib/tools/db2vwddd.jar:/home/db2inst/sqllib/tools/db2cmn.jar:/home/db2inst/sqllib/java/db2jcc.jar:

#PATH=./usr/bin:/usr/bins/etc:/usr/sbin:/usr/ucb:/usr/bin/
X11:/sbin:/home/db2inst/sqllib/bin:/home/db2inst/sqllib/bin:/home/db2inst/sqllib/bin:/home/db2inst/sqllib/bin:
```

Procedure:

To publish warehouse metadata:

1. Prepare the environment.
2. Select the metadata to publish.
3. Update published metadata.
4. Schedule regular updates of published metadata.

Related concepts:

- “How Data Warehouse Center metadata is displayed in the information catalog” in the *Information Catalog Center Administration Guide*

Exporting and importing metadata in the Data Warehouse Center

- “Maintenance for published objects in the Data Warehouse Center” in the *Information Catalog Center Administration Guide*
- “Regular updates to Data Warehouse Center metadata” in the *Information Catalog Center Administration Guide*

Related tasks:

- “Preparing to publish Data Warehouse Center metadata” in the *Information Catalog Center Administration Guide*

Related reference:

- “Metadata Mappings between the Information Catalog Center and the Data Warehouse Center” in the *Information Catalog Center Administration Guide*

Increasing the application heapsize for publishing Data Warehouse Center metadata

When you are publishing a large amount of Data Warehouse Center metadata, you might exceed the heapsize of the information catalog database. To correct this problem, you must increase the application heapsize.

Procedure:

To increase the application heapsize, enter the following command at a DB2 command prompt:

```
update db cfg for dbname using applheapsz 1024
```

Where *dbname* is the name of the information catalog database.

MQSeries

This section describes how to set up MQSeries in the Data Warehouse Center. MQSeries is a family of IBM licensed programs that provides message queuing services.

MQSeries data

With the Data Warehouse Center you can access data from an MQSeries® message queue as a DB2® database view. A wizard is provided to help you create a DB2 table function and the DB2 view through which you can access the data. Each MQSeries message is treated as a delimited string, which is parsed according to your specification and returned as a result row. In addition, MQSeries messages that are XML documents can be accessed as a warehouse source. Using the Data Warehouse Center, you can import metadata from an MQSeries message queue and a DB2 XML Extender Document Access Definition (DAD) file.

The following warehouse objects are added to the Warehouse tree when the import operation is complete.

- A subject area named MQSeries and XML
- A process named MQSeries and XML
- A user-defined program group named MQSeries and XML
- Definitions of all warehouse target tables described in the DAD file
- The *ServiceName.DAD file base name.Warehouse target Name* step

Exporting and importing metadata in the Data Warehouse Center

- The *ServiceName.DAD file base name* program template

Related concepts:

- “MQSeries stored procedure” on page 178
- “Error logging for MQSeries” on page 179

Related tasks:

- “Creating views for MQSeries messages” on page 176
- “Importing MQSeries messages and XML metadata” on page 176

Creating views for MQSeries messages

You can create a view to access data from a MQSeries message queue. When you catalog a warehouse source database, the database alias is cataloged on the warehouse agent site. However, when you create MQSeries and XML views, the Data Warehouse Center assumes that the database alias is also defined on the client workstation and will attempt to connect to it using the warehouse source database user ID and password. If the connection is successful, the wizard opens and you can create the view. If a connection is not successful, a warning message is displayed, and you must either catalog a database or choose a different database alias in the wizard.

Prerequisites:

The following programs are required to create views for MQSeries data in the Data Warehouse Center:

- DB2 Universal Database Version 7.2 or later
- DB2 Warehouse Manager Version 7.2 or later
- MQSeries Support

Procedure:

To create a view for MQSeries messages:

1. From the Data Warehouse Center window, expand the **Warehouse Sources** tree.
2. Expand the warehouse source that is to contain the view.
3. Right-click the **Views** folder, and click **Create for MQSeries messages**.

The MQSeries wizard opens. When you complete the wizard, a new view is created in the Data Warehouse Center. When you select the view, the MQSeries queue is accessed and each message is parsed as a delimited string according to your specifications in the wizard.

Related reference:

- “Parameters for a DB2 for z/OS Utility” on page 211

Importing MQSeries messages and XML metadata

You can import MQSeries messages and XML metadata into the Data Warehouse Center.

Prerequisites:

- The following programs must be installed:
 - DB2 Universal Database, Version 7.2 or later

Exporting and importing metadata in the Data Warehouse Center

- DB2 XML Extender, Version 7.2 or later
- MQSeries Support
- You must define the warehouse target.
- The import process will fail if the target tables exist with primary or foreign keys. You must manually delete the definitions of these keys in the Data Warehouse Center before you import MQSeries and XML metadata.

Procedure:

To import MQSeries metadata into the Data Warehouse Center:

1. Prepare the warehouse target database:
 - a. Define the warehouse target and register and enable the transformers.
 - b. Enable the warehouse target for DB2 XML Extender.
 - c. Create an XML Extender Data Access Definition (DAD) file to tell the Data Warehouse Center how to map the contents of the XML document to warehouse tables. Enable an XML collection using the DAD file for the database.
2. Right-click **Warehouse**, and click **Import Metadata** → **MQSeries** to open the **Import Metadata** window.
3. In the **AMI service** field, type the service point that a message is sent to or retrieved from.
4. In the **AMI policy** field, type the policy that the messaging system will use to perform the operation.
5. In the **DAD file** field, type the name of the DB2 XML Extender DAD file, or search for a file to select by clicking the ellipsis (...). This file must be local.
6. In the **Warehouse Target** field, select the name of the warehouse target where the step will run.
7. In the **Schema** field, type the name of a schema to qualify the table names in the DAD file that do not have a qualifier. The default schema is the logon user ID of the warehouse target that you selected previously.
8. Choose a **Target Option**:
 - If you want the step to replace the target table contents at run time, click **Replace table contents**.
 - If you want the step to append to the target table contents at run time, click **Append table contents**.
9. Click **OK**.

The Import Metadata window closes.

If the warehouse target agent site is remote, you must change the step parameter:

1. Right-click the step, and click **Properties** to open the Properties notebook for the step.
2. Click the **Parameters** tab.
3. Change the name of the DAD file parameter to the name of the same DAD file on the remote warehouse target agent site.
4. Make sure the **Agent Site** field on the Processing Options page contains the correct agent site.

Related concepts:

- “MQSeries data” on page 175
- “MQSeries stored procedure” on page 178

Exporting and importing metadata in the Data Warehouse Center

- “Error logging for MQSeries” on page 179

Related tasks:

- “Creating views for MQSeries messages” on page 176

Related reference:

- “Error return codes for the MQXMLXF stored procedure” on page 178

MQSeries stored procedure

The MQSeries[®] and XML stored procedure is called MQXMLXF, and is included with DB2[®] Data Warehouse Center Version 8.1 for Windows[®] NT and UNIX[®]. The step that is created when you import MQSeries and XML metadata will run the stored procedure. Its parameters are described in the following table.

Table 34. Parameters for the stored procedure

Parameter	Values
MQSeries ServiceName	The name of the service point that a message is sent to or retrieved from.
MQSeries PolicyName	The name of the policy that the messaging system will use to perform the operation.
DAD file name	The name of the DB2 XML Extender DAD file
TargetTableList	List of target tables of the step, separated by commas
Option	REPLACE or APPEND
RUN ID	Step edition number (for logging purposes)

Note: The data type for all of the parameters in this table is CHARACTER.

If the Option parameter has a value of REPLACE, the stored procedure deletes all rows from the target tables. The stored procedure also calls the DB2 XML Extender stored procedure to populate the target tables for all existing MQSeries messages.

Related concepts:

- “MQSeries data” on page 175
- “Error logging for MQSeries” on page 179

Related tasks:

- “Creating views for MQSeries messages” on page 176
- “Importing MQSeries messages and XML metadata” on page 176

Related reference:

- “Error return codes for the MQXMLXF stored procedure” on page 178

Error return codes for the MQXMLXF stored procedure

When the stored procedure runs, it might return error code SQLCODE -443 and SQLSTATE 38600.

The following table shows the diagnostic texts for error codes returned by the MQXMLXF stored procedure.

Table 35. Error return codes

Error number	Description
AMIRC=xxxxx;<log file name>	xxxxx is the return code from the AMI layer. See the MQSeries documentation for more details. <log file name> indicates the location of the log file.
XMLRC=xxxxx;<log file name>	xxxxx is the return code from the DB2 XML Extender. See the DB2 XML Extender documentation for descriptions of the return codes. <log file name> indicates the location of the log file.
SQLCODE=xxxxx;<log file name>	xxxxx is the nonzero SQLCODE returned when an SQL request is performed. <log file name> indicates the location of the log file.

Related concepts:

- “MQSeries data” on page 175
- “MQSeries stored procedure” on page 178
- “Error logging for MQSeries” on page 179

Related tasks:

- “Importing MQSeries messages and XML metadata” on page 176

Error logging for MQSeries

The Data Warehouse Center stores all diagnostic information in a log file when the MQXMLXF stored procedure runs. The name of the log file is mqxfnnnnnnnnn.log, where *nnnnnnnn* is the run ID that was passed to the stored procedure. The Data Warehouse Center creates the file in the directory that is specified by the VWS_LOGGING environment variable. If this environment is not defined, the log file will be created in the temporary directory.

To make the VWS_LOGGING environment variable visible to the stored procedure on a UNIX[®] system, add VWS_LOGGING to the DB2ENVLIST environment variable using the **db2set** command before the **db2start** command.

The following example shows an environment command:

```
db2set DB2ENVLIST="AMT_DATA_PATH VWS_LOGGING"
```

The log file is deleted if the step runs successfully.

Related concepts:

- “MQSeries data” on page 175
- “MQSeries stored procedure” on page 178

Related tasks:

- “Creating views for MQSeries messages” on page 176
- “Importing MQSeries messages and XML metadata” on page 176

Related reference:

- “Error return codes for the MQXMLXF stored procedure” on page 178

Importing ERwin metadata

The following topics describe how to extract ERwin metadata into the Data Warehouse Center and Information Catalog Center. If you use ERwin 4.0 or later, you extract and import the metadata from an XML file. If you use ERwin 3.5.2, you extract ERwin metadata from an .ER1 file to a .tag file, then import the .tag file.

IBM ERwin 3.5.2 metadata extract program

The ERwin Metadata Extract program extracts all objects such as databases, tables, and columns that are stored in the input ER1 file, and writes the metadata model to a Data Warehouse Center or an Information Catalog Center tag language file. The logical model for the Information Catalog Center, which consists of entities and attributes, is also extracted and created. The relevant relationship tags between objects, such as between databases and tables and between tables and entities, are also created. For tables without a database, a default database named DATABASE is created. For tables without a schema, a default schema of USERID is used. For a model name, the ER1 file name is used.

The ERwin Metadata Extract program supports all ER1 models with relational databases, including DB2[®], Informix[®], Oracle[®], Sybase, ODBC data sources, and Microsoft[®] SQL Server.

For ERwin 4.0 or later, use the db2erwinimport command to import ERwin data from an XML file.

Related concepts:

- “IBM ERwin Metadata Extract program errors” on page 186

Related tasks:

- “iwh2imp2 command syntax for ERwin 3.5.2 to import metadata into the Data Warehouse Center” on page 185
- “Running the ERwin metadata import commands” on page 180
- “Merging ERwin metadata with existing database data” on page 185
- “Creating tag language files for the IBM ERwin Metadata Extract program” on page 184

Related reference:

- “Mapping ERwin Version 3.5.2 object attributes to Data Warehouse Center tags” on page 236
- “db2erwinimport command for ERwin 4.0 or later” on page 181
- “Mapping ERwin Version 4.0 and later object attributes to Data Warehouse Center tags” on page 234

Running the ERwin metadata import commands

The command that you run depends on which version of ERwin you have.

Prerequisites:

The following software is required to import ERwin 3.5.2 metadata:

- Windows NT 4.0 or later
- ERwin 3.5.2 with Service Pack 3 Build 466 or later

Exporting and importing metadata in the Data Warehouse Center

The following software is required to import ERwin 4.0 metadata:

- AIX, Solaris Operating Environment, or Windows NT 4.0 or later
- ERwin 4.0 or later

Procedure:

If you use ERwin 3.5.2, perform the following steps to import ERwin metadata:

1. To generate a .tag file from the ER1 file, enter the command **flgerwin**.
2. To import the .tag file into the Data Warehouse Center, enter the command **iwh2imp2**.

Alternatively, you can import the ER1 file through the Data Warehouse Center.

If you use ERwin 4.0 or later, perform the following steps to import ERwin metadata:

1. In ERwin, to save the model you want to import to the Data Warehouse Center as an XML file, click **File** → **Save as** and select XML as the file type.
2. To import the XML file into the Data Warehouse Center or the Information Catalog Center, enter the command **db2erwinimport**.

Related tasks:

- “Promoting a step: Data Warehouse Center help”
- “Steps and tasks: Data Warehouse Center help”

db2erwinimport command for ERwin 4.0 or later

This command imports data from an ERwin 4.0 or later XML file into a data warehouse or an information catalog. You do not need ERwin installed on the same system as the DB2 client, but the system that runs the command must be able to read the XML input file that ERwin generated. The command runs on DB2 clients on the following operating systems:

- AIX
- Solaris Operating Environment
- Windows

Syntax to import metadata from ERwin 4.0 or later:

```
db2erwinimport -x xmlfile [-t tracefile] [-dwc <m><s>] [-dwcdb ctrlldb]  
[-dwcs dwcprefix] [-dwcuser dwcuser] [-dwcp dwcpass] [-icm <m | u | r>]  
[-icmdb icmdb] [-icms icmschema] [-icmu ucmuser] [-icmp icmpass]
```

Where:

-x *xmlfile*

The full or relative path and name of the XML file that ERwin created.

-t *tracefile*

The full or relative path and name of the trace file. The directory names that you specify must exist before you run the command. The trace file provides information to IBM in case of a problem. Do not specify this option unless you encounter a problem that you cannot resolve.

-dwc <m><s>

Specifies that ERwin metadata be imported into the Data Warehouse Center.

Exporting and importing metadata in the Data Warehouse Center

- | **m** Merges added and updated objects. If you do not specify this
- | option, the default value *m* is used.
- |
- | **s** Creates a star schema using the imported metadata.
- |
- | You can import metadata into the Data Warehouse Center and an
- | information catalog with a single command by specifying **-dwc** and **-icm**
- | with their associated options.
- |
- | **-dwcd** *ctrlldb*
- | The name of the Data Warehouse Center control database that is cataloged
- | on the client system. If you do not specify this option, the default control
- | database, DWCTRLDB, is used.
- |
- | **-dwcs** *dwcprefix*
- | The prefix or schema of the Data Warehouse Center control database. If
- | you do not specify this option, the default prefix, IWH, is used.
- |
- | **-dwcu** *dwcuser*
- | An administrator ID for the Data Warehouse Center. Only administrators
- | have import authority.
- |
- | **-dwcp** *dwcpassword*
- | The password associated with the administrator ID.
- |
- | **-icm** *<m | u | r>*
- | Specifies that metadata be imported to the Information Catalog Manager.
- |
- | **m** Merges added and updated objects with objects in the information
- | catalog.
- |
- | **u** Updates existing objects and adds new objects to the information
- | catalog. Deletes objects that are in the information catalog, but not
- | in the imported metadata.
- |
- | **r** Replaces existing objects in the metadata and deletes objects that
- | are in the information catalog but not in the imported metadata.
- |
- | You can import metadata into an information catalog and into the Data
- | Warehouse Center with a single command by specifying the **-icm** and
- | **-dwc** parameters with their associated options.
- |
- | **-icmd** *icmdb*
- | The name of the information catalog to use. The default information
- | catalog, ICMDB, is used if you do not specify this option.
- |
- | **-icms** *icmschema*
- | The schema or catalog name of the information catalog to use. The default
- | schema, ICM, is used if you do not specify this option.
- |
- | **-icmu** *icmuser*
- | The administrator ID for Information Catalog Manager. The ID is a DB2
- | authorization ID for the system that contains the information catalog
- | database.
- |
- | **-icmp** *icmpassword*
- | The password associated with the Information Catalog Manager
- | administrator ID.

Table 36. Return codes for the *db2erwinimport* command

Return code	Explanation
4	Error encountered parsing input parameters.

Table 36. Return codes for the db2erwinimport command (continued)

Return code	Explanation
8	Required parameters are not specified.
12	I/O error while reading XML file or writing trace file.
16	Uncaught exception. Turn on trace for information on the cause of the exception.
20	Null pointer exception. Turn on trace for more information on the cause of this exception.
100	Environment error, VWS_LOGGING or VWS_TEMPLATES environment variables not set.
101	Model error. Import into DWC specified, but ERwin model is logical only. Or unable to determine data type of attribute. Turn on trace for more information.
102	Unable to create tag file for DWC import.
105	Exception occurred parsing XML file, possibly wrong XML version. Turn on trace for more information.
106	Database server of the ERwin physical model is not supported. See below for TargetServer's for which the ERwin bridge supports.
107	Tag import to DWC failed. See the tag import log for more information.
108	Information Catalog connection failed. Check the catalog name, user ID and password values.
109	ICMException occurred while importing to Information catalog. Turn on trace for more information on the cause of the exception.
110	Data Warehouse Exception occurred during DWC import. Turn on trace for more information on the cause of the exception.

Related concepts:

- "IBM ERwin 3.5.2 metadata extract program" on page 180

Related tasks:

- "iwh2imp2 command syntax for ERwin 3.5.2 to import metadata into the Data Warehouse Center" on page 185
- "Running the ERwin metadata import commands" on page 180
- "Merging ERwin metadata with existing database data" on page 185
- "Creating tag language files for the IBM ERwin Metadata Extract program" on page 184

Related reference:

- "Mapping ERwin Version 3.5.2 object attributes to Data Warehouse Center tags" on page 236
- "ERwin to Information Catalog Center mapping" in the *Information Catalog Center Administration Guide*
- "Metadata mappings between ERwin Version 4.0 object attributes and Information Catalog Center properties" in the *Information Catalog Center Administration Guide*
- "Mapping ERwin Version 4.0 and later object attributes to Data Warehouse Center tags" on page 234

Creating tag language files for the IBM ERwin Metadata Extract program

You can create a tag language file for the Data Warehouse Center or Information Catalog Center by using the ERwin Metadata Extract program

Procedure:

To use the IBM ERwin Metadata Extract program to create a tag language file from the command line, enter the flgerwin command.

Syntax for creating a tag language file

```

flgerwin inputFile.er1 outputFile.tag [-dwc -m -starschema] [-icm [-m [-u [-a [-d]]]]]
    
```

Where *inputFile.er1* is the name of your input file, and *outputFile.tag* is the name of your output tag language file.

- dwc** Default. Creates a Data Warehouse Center tag language file. Optional parameters are -m and -starschema.
- icm** Creates an Information Catalog Center tag language file. Optional parameters are -m, -u, -a, and -d. You can specify only one parameter for -icm. If you do not specify any parameters, then -icm runs as though -m was specified.
- starschema** Creates an ERwin model star schema tag language file.
- m** Default. Specifies the action on the object as MERGE.
- u** Optional. Specifies the action on the object as UPDATE.
- a** Optional. Specifies the action on the object as ADD.
- d** Optional. Specifies the action on the object as DELETE.

The ERwin Metadata Extract program creates a tag language file that defines the warehouse targets when it is imported into the warehouse. The warehouse targets are defined in the Data Warehouse Center as logical entities and might not exist or might be empty.

Related concepts:

- “IBM ERwin 3.5.2 metadata extract program” on page 180

Related tasks:

- “iwh2imp2 command syntax for ERwin 3.5.2 to import metadata into the Data Warehouse Center” on page 185

Related reference:

- “Mapping ERwin Version 3.5.2 object attributes to Data Warehouse Center tags” on page 236

Merging ERwin metadata with existing database data

When you import ERwin metadata, the Data Warehouse Center defines the database and tables to the warehouse, but for the warehouse to access the database, you must supply the correct user ID and password.

Procedure:

To merge metadata with existing database data:

1. Click **Properties** → **Database** → **Userid**.
2. Change the Data Warehouse Center user ID and password to match the merged database user ID and password.
3. Click **OK** to save your changes and close the window.

Returning the ER1 file to a writable state

The ERwin Metadata Extract program saves the ER1 file in a read-only state if the file is being used in a current ERwin session or if some error condition was detected. You might receive an abnormal program termination error message if the ER1 file is in a read-only state. The ERwin Metadata Extract program displays the table name that it is currently processing. You will receive an informational message when the metadata extract program finishes processing.

Procedure:

The input ER1 file must be in a writable state. After the metadata extract program runs, the ER1 file becomes read-only. To change the file to read/write mode, use a command such as the following example:

```
attrib -r erwinsimplemode.er1
```

where `erwinsimplemode.er1` is the name of the ERwin flat file.

iwh2imp2 command syntax for ERwin 3.5.2 to import metadata into the Data Warehouse Center

Use the following command to import metadata from an ERwin 3.5.2 .tag file into the Data Warehouse Center. **Restriction:** If you want to change the metadata that you import into the Data Warehouse Center, you must edit the tag language file manually, and import the file using the Data Warehouse Center import capabilities.

```
iwh2imp2 tag-filename log-pathname target-control-db userid password
```

tag-filename

The full path and file name of the tag language file.

log-pathname

The full path name of the log file.

target-control-db

The name of the target database for the import.

userid The user ID used to access the control database.

password

The password used to access the control database.

Exporting and importing metadata in the Data Warehouse Center

Related concepts:

- “IBM ERwin 3.5.2 metadata extract program” on page 180

Related tasks:

- “Creating tag language files for the IBM ERwin Metadata Extract program” on page 184
- “Importing an ERwin tag language file: Data Warehouse Center help”

Changing the tag language file to change the DB2 database definition to a source in the Data Warehouse Center

By default, the ERwin Metadata Extract program generates tag language files that define databases as warehouse targets. If you want to import these databases as warehouse sources, you must change the tag language file that is generated by the extract program. You can change the tag language file by running the extract program from the command line, editing the tag language file, and importing the tag language file into the warehouse.

To change a DB2 database definition so that it is a source in the Data Warehouse Center, you can change the tag language file.

Procedure:

To change the tag language file:

- Change the ISWH tag from ISWH(Y) to ISWH(N) for each database that you want as a source.
- Change the relationship tag from :RELTYPE.TYPE(LINK) SOURCETYPE(SCGTARIR) TARGETTYPE(DATABASE) to :RELTYPE.TYPE(LINK) SOURCETYPE(SCGSR CIR) TARGETTYPE(DATABASE) for each database that you want as a source.

Related tasks:

- “iwh2imp2 command syntax for ERwin 3.5.2 to import metadata into the Data Warehouse Center” on page 185

IBM ERwin Metadata Extract program errors

If you receive an error message, look for the message and an explanation of how to resolve the error in this topic.

Missing ER1 input file or tag output file.

The ERwin Metadata Extract program requires two parameters in a specific order. The first parameter is the name of the ER1 file. The second parameter is the name of the tag language output file. If you specify the name of an existing tag language file, the file is overwritten.

Windows® system abnormal program termination.

The input ER1 file is probably in a read-only state. This can happen if a problem occurred when you saved the ER1 file, and the ERwin Metadata Extract program put the file in read-only state. Enter the following command:

```
attrib -r inputFile.er1
```

in a command shell to change the state of the ER1 file to read/write.

Exporting and importing metadata in the Data Warehouse Center

Tag language file *file name* could not be opened.

Check if any system problems exist that might prevent a file from being created or opened on the current drive.

Path to template files not found.

The environment variable VWS_TEMPLATES is not set. Verify that the Data Warehouse Center is installed.

Unsupported server version: *Version*

The input ER1 file that you are trying to extract from is stored on a target server that is not supported by the program. Start the ERwin Metadata Extract program, open the ER1 file, then click **Server** —> **Target Server** and the appropriate version. Save the ER1 file.

Unknown ERwAPI error.

An ERwin API error occurred, and the program could not obtain more information on the error. Make sure that ERwin 3.5.2 is installed. You must register the ERwin API.

To register the ERwin API, run the following command from the directory where your ERwin program files are installed:

```
regsvr32 er2api32.dll
```

You can start the extract program from the Data Warehouse Center, or by issuing the flgerwin command from a command line.

Extract program error: *error message*

Check the error message and take action as appropriate. Most likely, this is an internal extract program error and the problem needs to be reported to IBM® Software Support.

Unknown extract program error.

An unknown error occurred. Most likely, this is an internal error and the problem needs to be reported to IBM Software Support.

Extract program terminated due to error(s).

An error occurred that prevents the completion of the extract program. Refer to additional error messages to solve the problem or contact IBM Software Support.

Use a unique constraint name.

Use the automatically generated constraint name to ensure that the constraint name is unique.

Duplicate column found. Column will not be extracted.

This message is informational and does not affect the successful completion of the extract program. This message is displayed when the physical name of a foreign key is the same as the physical name of a column in the table that is currently being processed.

The object type "COLUMN" identified by "DBNAME(____) OWNER(____) TABLE(____) COLUMNS(____)" is defined twice in the tag language file.

This message is informational, and your import was completed successfully. You might receive this message if an entity has foreign keys with the same name, or an entity has similarly named columns that were affected by truncation, or other similar circumstances. Check your model for duplicate column names, and make changes as appropriate.

Related concepts:

Exporting and importing metadata in the Data Warehouse Center

- “IBM ERwin 3.5.2 metadata extract program” on page 180

Related tasks:

- “Running the ERwin metadata import commands” on page 180

Related reference:

- “db2erwinimport command for ERwin 4.0 or later” on page 181

Chapter 15. Extending the Data Warehouse Center

You can extend the function of the Data Warehouse Center with the following programs:

- User-defined programs
- Microsoft OLE DB and Data Transaction Services (DTS)

User-defined programs

You can use user-defined programs to use the best data warehouse software for your needs, while providing a single point-of-control to administer the warehouse. The Data Warehouse Center will start an application that you define as a user-defined program at a scheduled time.

For example, if you have a data cleansing program that you want to use on your warehouse tables, you can define the data cleansing program as a user-defined program, and run a step for that program that starts after a step that populates the warehouse tables.

What is a program group?

A program group is a logical group that contains related user-defined programs. You must create a program group before you can define a user-defined program to the Data Warehouse Center.

What is a user-defined program?

A *user-defined program* is a program that you create or a program that is created by a vendor. You define a user-defined program to the Data Warehouse Center so that one or more steps can use the program for processing. If you are defining a stored procedure, you must ensure that the stored procedure exists in the target database. If you are defining any other type of program, you must install the program on one or more Data Warehouse Center agent sites before you define the program to the Data Warehouse Center.

After you define a user-defined program to the Data Warehouse Center, the program definition is available for use as a step in the Process Model window.

When a user-defined program has run, the System Message and the Comment will be written to the warehouse log file. These messages are now visible from the Work in Progress window.

Related concepts:

- “Example: Specifying pre-defined parameters for a user-defined program” on page 193
- “Parameters for user-defined programs” on page 191
- “Agent sites for user-defined programs” on page 190
- “Status information for user-defined programs” on page 194
- “Example: Passing parameters to a user-defined program” on page 192

Related tasks:

- “Enabling a user-defined program (Windows)” on page 192
- “Using a user-defined program in the Data Warehouse Center” on page 190
- “Defining a step that runs a user-defined program: Data Warehouse Center help”
- “Define a user-defined program to the Data Warehouse Center: Data Warehouse Center help”
- “Defining a user-defined program to the Data Warehouse Center: Data Warehouse Center help”

Related reference:

- “User-defined programs” on page 97

Using a user-defined program in the Data Warehouse Center

This topic explains how to define a user-defined program in the Data Warehouse Center.

When you define a step that runs a user-defined program, you can change the parameter values that are defined for the program. If you change the parameter values for the program, the changes affect only the instance of the program that is being used in the step. The changes do not affect the original program definition.

To permanently change the parameter values of a user-defined program, you must edit the program definition.

Example: You define a step that uses the user-defined program that you defined in the previous section. The step has no source. Because you are using the file to be found as a source for the next step in the sequence, you define the file as a target for this step. You then define a load step that uses the file as a source. The load step loads the file into a database.

Prerequisites:

If your user-defined program uses tokens for a source or target, you must link this step to the source or target.

Procedure:

To use a user-defined program in the Data Warehouse Center:

1. In the Data Warehouse Center, define a program group
2. Define a user-defined program to the Data Warehouse Center.
3. Define a step that uses the user-defined program.
4. Test the step that uses the user-defined program.

Agent sites for user-defined programs

On the Agent Sites page of the Program notebook, you must select the agent site on which the program is installed.

If you specified a user ID and password when you defined the agent site, the program will run as a user process. If you did not specify a user ID and password, the program will run however the warehouse agent was defined.

You can run some programs as user processes and other programs as system processes on the same workstation. To do this, define two agent sites on the workstation: one that has a user ID and password, and one that does not.

Related concepts:

- “Agent site configurations” on page 17

Related tasks:

- “Defining an agent site” on page 18

Parameters for user-defined programs

You can use predefined tokens for some parameters. The Data Warehouse Center substitutes values for the tokens at run time. For example, there is a token for the database name of the target resource for a step, &TDB. If you include this token in your parameter list, the Data Warehouse Center provides the name of the database defined in the notebook of the warehouse target that contains the target table that is linked to the step. Tokens allow you to change the values that are passed depending on which step uses the program.

Procedure:

To add a system parameter or system token to a program definition, open the Define User Program notebook to the Parameters page and add the system parameters or system tokens that you want to use.

Related concepts:

- “Example: Specifying pre-defined parameters for a user-defined program” on page 193
- “Example: Passing parameters to a user-defined program” on page 192

Related reference:

- “User-defined programs” on page 97

Writing your own program for use with the Data Warehouse Center

Programs that you write for use with the Data Warehouse Center

You can write programs in any language that supports one of the following program types: executable, command program, dynamic link library, or stored procedure.

If the user-defined program is not of type stored procedure, it must reside on the agent site. The warehouse agent starts the program at the scheduled time. On Windows[®] NT, Windows 2000, or Windows XP, the agent runs as a system process by default, which means that the program cannot access resources or programs that require a user ID. Also, any environment variables that the program needs to access must be system variables.

If you write user-defined programs using Object REXX for Windows, you must enable the agent to run under Windows NT[®], Windows 2000, or Windows XP as a user process

Stored procedures must run on target warehouses that the agent site can access.

Enabling a user-defined program (Windows)

If you write user-defined programs using Object REXX for Windows, you must enable the agent to run under Windows NT, Windows 2000, or Windows XP as a user process

Procedure:

To enable a user-defined program to run under Windows:

1. Define the warehouse agent or server service as a system process that can interact with the Windows desktop:
 - a. Select the agent or server service from the **Service** list.
 - b. Click **Startup**.
 - c. Click the account.
 - d. Specify a valid user ID and password.
2. Initialize the Object REXX environment before the warehouse agent or server starts the program. You can initialize the environment by running any Object REXX program from the command line.
3. If your Object REXX program issues a DB2 CONNECT statement, verify that the statement includes the user ID and password, as in the following example:

```
DB2 CONNECT TO testdb USER db2admin USING db2pass
```

If the user-defined program has a program type of stored procedure, the program must reside in the warehouse database that is associated with the step that runs the program.

Related concepts:

- “Example: Specifying pre-defined parameters for a user-defined program” on page 193
- “Parameters for user-defined programs” on page 191
- “Agent sites for user-defined programs” on page 190
- “What is a user-defined program?” on page 189
- “Status information for user-defined programs” on page 194
- “Example: Passing parameters to a user-defined program” on page 192

Related tasks:

- “Using a user-defined program in the Data Warehouse Center” on page 190

Related reference:

- “User-defined programs” on page 97

Example: Passing parameters to a user-defined program

At run time, the Data Warehouse Center generates a command line parameter list that it passes as input to your user-defined program. Whenever possible, test your program from the command line before using it in a step.

Example: You write a user-defined program that checks for a file at regular intervals on a Windows workstation. The program uses the following parameters:

- File name
- Polling interval

- Timeout interval

The program gets the parameters as shown in the following example:

```
char *remoteFile = argv[1];
int interval = 5; // check every 5 minutes
int timeout = 60; // time-out period in minutes

if( (argc==3) || (argc==4) )
{
    if( NumStrValidation( argv[2] ) ) // get polling interval
    {
        interval = atoi( argv[2] );
        if( interval == 0 )
            interval = 5; // used the original one
    }
    else
    {
        trace << "Invalid value for polling interval\n";
        flush(trace);
        return PARAMETER_ERROR;
    }

    if( argc==4 ) // get timeout period
    {
        if( NumStrValidation( argv[3] ) )
        {
            timeout= atoi( argv[3] );
            if( timeout == 0 )
                timeout = 480; // used the original one
        }
        else
        {
            trace << "Invalid value for timeout period\n";
            flush(trace);
            return PARAMETER_ERROR;
        }
    }
}
}
```

Related concepts:

- “Example: Specifying pre-defined parameters for a user-defined program” on page 193
- “Parameters for user-defined programs” on page 191

Related tasks:

- “Enabling a user-defined program (Windows)” on page 192
- “Using a user-defined program in the Data Warehouse Center” on page 190

Related reference:

- “User-defined programs” on page 97

Example: Specifying pre-defined parameters for a user-defined program

You are defining a user-defined program that checks for a file at regular intervals on a Windows workstation. You intend to use this program to find a file that another step will load into a database.

The program uses the following parameters:

- File name

Extending the Data Warehouse Center

- Polling interval
- Timeout interval

You use the Warehouse target file name system parameter (&TTBN) to represent the file name. You define your own parameters for the polling interval and timeout interval.

Related concepts:

- “Parameters for user-defined programs” on page 191
- “Example: Passing parameters to a user-defined program” on page 192

Related reference:

- “User-defined programs” on page 97

Status information for user-defined programs

After your program runs, it must return a return code to the step that uses the program. The return code must be a positive integer. If your program does not return a return code, the step that uses the program might fail. The Data Warehouse Center displays the return code in the **Error RC2** field of the Log Details window when the value of **Error RC1** is 8410. If the value of **Error RC2** is 0, then the program ran successfully without errors.

Your program can return the following additional status information to the Data Warehouse Center:

- Another return code, which can be the same as or different from the code that is returned by the user-defined program.
- A warning flag that indicates an SQL warning code, or that the user-defined program found no data in the source table. When this flag is set, the step that uses this program will have a status of **Warning** in the Operations Work in Progress window.
- A message, which the Data Warehouse Center will display in the **System Message** field of the Log Viewer Details window.
- The number of rows of data processed by the user-defined program, which the Data Warehouse Center displays in the Log Viewer Details window for the step.
- The number of bytes of data processed by the user-defined program, which the Data Warehouse Center displays in the Log Viewer Details window for the step.
- The SQLSTATE return code, which the Data Warehouse Center displays in the SQL state field of the Log Viewer Details window.

The warehouse agent transfers the additional status information to the warehouse server.

Transferring the information to the Data Warehouse Center: To transfer the additional status information to the warehouse agent, your program must create a file, called a *feedback file*, that contains the additional status information. The path and file name for the feedback file is specified in the VWP_LOG environment variable, which the warehouse agent sets before it calls the warehouse program. Your program must read the value of this variable, using the appropriate system call, to create the feedback file. After the user-defined program finishes running, the warehouse agent checks whether the feedback file exists. If it exists, the agent processes the file. Otherwise, the warehouse agent will do nothing. If the user-defined program cannot create the file, the program continues to run.

Format of the feedback file: Your user-defined program can write the additional status information to the feedback file in any order, but must use the following format to identify information. Enclose each item returned within the begin tag `<tag>` and end tag `</tag>` in the following list. Each begin tag must be followed by its end tag; you cannot include two begin tags in a row. For example, the following tag format is valid:

```
<RC>...</RC>...<MSG>...</MSG>
```

The following embedded tag format is not valid:

```
<RC>...<MSG>...</RC>...</MSG>
```

You can specify the following information in the feedback file:

Return code

`<RC>return code</RC>`, where *return code* is a positive integer.

Return code warning flag

`<WARNING>1</WARNING>` sets the return code warning flag to on. If this flag is on, the step's processing status in the Work in Progress window is Warning.

Data Warehouse Center system message

`<MSG>message text\n</MSG>`

message text

The text of one or more messages

`\n` The new line character. Include this character at the end of each message if there are multiple messages.

Comment

`<COMMENT>comment text</COMMENT>`, where *comment text* is the text of the comment.

Number of rows of data processed

`<ROWS>number of rows</ROWS>`, where *number of rows* is any positive integer.

Number of bytes processed

`<BYTES>number of bytes</BYTES>`, where *number of bytes* is any positive integer.

SQLSTATE

`<SQLSTATE>sqlstate string</SQLSTATE>`, where *sqlstate string* is any string whose length is greater than 0 and less than or equal to 5 digits.

The following example shows the feedback file.

```
<RC> 20</RC>
<ROWS>2345</ROWS>
<MSG>The parameter type is not correct</MSG>
<COMMENT> Supply the correct parameter type (PASSWORD
    NOTREQUIRED, GETPASSWORD, ENTERPASSWORD)</COMMENT>
<BYTES> 123456</BYTES>
<WARNING> 1</WARNING>
<SQLSTATE>12345</SQLSTATE>
```

How the feedback determines the step status: The return codes and step status for the user-defined program that are displayed in the log viewer varies depending on the following values set by the program:

Extending the Data Warehouse Center

- The value of the return code returned by the user-defined program
- Whether a feedback file exists
- The value of the return code in the feedback file
- Whether the warning flag is set on

The following table lists the possible combinations of these values and the results that they produce. In this table, the Step status column shows the step status as it is displayed in the Work in Progress window.

Table 37. Step status for user-defined programs based on certain conditions

Conditions	Step status	Values of Error RC1 and RC2
<ul style="list-style-type: none"> • The user-defined program return code is 0 • No feedback file exists 	Successful	RC1 = 0 RC2 = 0
<ul style="list-style-type: none"> • The user-defined program return code is 0 • A feedback file exists • The value of <RC> in the feedback file is 0 • <WARNING> is not set in the feedback file 	Successful	RC1 = 0 RC2 = 0
<ul style="list-style-type: none"> • The user-defined program return code is 0 • A feedback file exists • The value of <RC> in the feedback file is 0 • The value of <WARNING> in the feedback file is 1 	Warning	RC1 = 0 RC2 = 0
<ul style="list-style-type: none"> • The user-defined program return code is 0 • A feedback file exists • The value of <RC> in the feedback file is nonzero • <WARNING> is not set in the feedback file 	Failed	RC1 = 8410 (the user-defined program failed) RC2 = the value of <RC> in the feedback file
<ul style="list-style-type: none"> • The user-defined program return code is 0 • A feedback file exists • The value of <RC> in the feedback file is nonzero • The value of <WARNING> in the feedback file is 1 	Warning	RC1 = 0 RC2 = the value of <RC> in the feedback file
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • No feedback file exists 	Failed	RC1 = 8410 (the user-defined program failed) RC2 = the code returned by the user-defined program

Table 37. Step status for user-defined programs based on certain conditions (continued)

Conditions	Step status	Values of Error RC1 and RC2
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • A feedback file exists • The value of <RC> in the feedback file is 0 • <WARNING> is not set in the feedback file 	Successful	RC1 = 0 RC2 = 0
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • A feedback file exists • The value of <RC> in the feedback file is 0 • The value of <WARNING> in the feedback file is 1 	Warning	RC1 = 0 RC2 = 0
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • A feedback file exists • The value of <RC> in the feedback file is nonzero • <WARNING> is not set in the feedback file 	Failed	RC1 = 8410 (the user-defined program failed) RC2 = the code returned by the user-defined program
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • A feedback file exists • The value of <RC> in the feedback file is nonzero • The value of <WARNING> in the feedback file is 1 	Warning	RC1 = 0 RC2 = the value of <RC> in the feedback file

The Data Warehouse Center checks for the existence of a feedback file, regardless of whether the return code for the user-defined program is 0 or nonzero. Also, the value of <RC> in the feedback file is always displayed as the value of the **RC2** field of the Log Details window.

Microsoft OLE DB and Data Transaction Services support

You can use the Data Warehouse Center to access data from an OLE DB provider as a DB2 database view. You can use the OLE DB Assist wizard provided with Data Warehouse Center to create a DB2 OLE DB table function and the DB2 view through which you can access the data.

Microsoft DTS is installed with Microsoft SQL Server. All DTS tasks are stored in DTS packages that you can run and access using Microsoft OLE DB Provider for DTS Packages. Because you can access packages from DTS as OLE DB sources, you can also create views with the OLE DB Assist wizard for DTS packages the same way as for OLE DB data sources. When you access the view at run time, the DTS package executes, and the target table of the task in the DTS package becomes the created view.

Extending the Data Warehouse Center

After you create a view in the Data Warehouse Center, you can use it as you would any other view. For example, you can join a DB2 table with an OLE DB source in an SQL step. When you use the new view in an SQL step, the DTS provider is called and the DTS package runs.

Creating views for OLE DB table functions

You can create a view to access data from an OLE DB provider in the Data Warehouse Center.

Prerequisites:

You must have the following software installed before beginning this task.

- DB2 Universal Database for Windows NT Version 7.2 or later as the warehouse target database
- DB2 Warehouse Manager Version 7.2 or later

Restrictions:

- If the warehouse target database was created before Version 7.2, you must run the db2updv7 command after installing the DB2 Universal Database for Windows NT Version 7.2 or later.
- When you catalog a warehouse source database, the database alias is cataloged on the warehouse agent site. However, when you start the wizard, the Data Warehouse Center assumes that the database alias is also defined on the client workstation and will attempt to connect to it using the warehouse source database user ID and password. If the connection is successful, the wizard starts and you can create the view. If the connection is not successful, a warning message is displayed and you must either catalog or choose a different database alias in the wizard.
- When you enter the table name for the wizard, use the step name, which is shown on the Options page of the Workflow Properties notebook for the task.
- When you enter the table name for the wizard, use the step name, which is shown on the Options page of the Workflow Properties notebook for the task.

Procedure:

To create a view for an OLE DB table function:

1. From the Data Warehouse Center window, expand the **Warehouse Sources** tree.
2. Expand the warehouse source that is to contain the view.
3. Right-click the **Views** folder, and click **Create for OLE DB table function** to open the OLE DB Assist wizard. The wizard steps you through the task of creating a new view in the warehouse source database.

Related tasks:

- “Creating views for MQSeries messages” on page 176
- “Creating views for DTS packages” on page 198

Creating views for DTS packages

You can use DTS to import, export, and transform data between OLE DB sources and targets to build data warehouses and datamarts.

Prerequisites:

You must have the following software installed before beginning this task.

- DB2 Universal Database for Windows NT Version 7.2 or later as the warehouse target database
- DB2 Warehouse Manager Version 7.2 or later

Restrictions:

- To identify a specific table from a DTS package, you must select the **DSO rowset provider** check box on the Options page of the Workflow Properties window of the DataPumpTask that creates the target table. If you turn on multiple DSO rowset provider attributes, only the result of the first selected step is used. When a view is selected, the rowset of its target table is returned and all other rowsets that you create in subsequent steps are ignored.
- The DTS package connection string has the same syntax as the dtsrun command.

Procedure:

To create a view for a DTS package:

1. From the Data Warehouse Center window, expand the **Warehouse Sources** tree.
2. Expand the warehouse source that is to contain the view.
3. Right-click the **Views** folder, and click **Microsoft OLE DB Provider for DTS Packages**.

The OLE DB Assist wizard opens. The wizard steps you through the task of creating a new view in the warehouse source database.

For more information about DTS, see the Microsoft Platform SDK 2000 documentation, which includes a detailed explanation of how to build the provider string that the wizard needs to connect to the DTS provider.

Related tasks:

- “Creating views for MQSeries messages” on page 176
- “Creating views for OLE DB table functions” on page 198

Extending the Data Warehouse Center

Chapter 16. Creating a star schema from within the Data Warehouse Center

This chapter describes how to use the Data Warehouse Center to create a star schema.

Star schemas

You can use the star schema in the DB2[®] OLAP Integration Server to define the cubes necessary to support OLAP customers. A *cube* is a set of data and metadata that defines a multidimensional database.

You populate a star schema within the Data Warehouse Center with cleansed data before you use the data to build a cube.

An OLAP *model* is a logical structure that describes how you plan to measure your business. The model takes the form of a star schema. A *star schema* is a specialized design that consists of multiple dimension tables, which describe aspects of a business, and one fact table, which contains the facts about the business. For example, if you have a mail-order business selling books, some dimension tables are customers, books, catalogs, and fiscal years. The fact table contains information about the books that are ordered from each catalog by each customer during the fiscal year. A star schema that is defined within the Data Warehouse Center is called a *warehouse schema*.

The following table describes the tasks involved in creating the warehouse schema and loading the resulting cube with data using the Data Warehouse Center and the DB2 OLAP Integration Server.

Table 38. Tasks and the product or component you use to perform each task

Task	Product or component to use
Define warehouse targets of relational data that you will use as sources for an OLAP cube.	Data Warehouse Center
Define the warehouse schema.	Data Warehouse Center
Export the warehouse schema to the DB2 OLAP Integration Server. The warehouse schema is used as a model in the DB2 OLAP Integration Server.	Data Warehouse Center
Use the model (warehouse schema) to define hierarchies for the model.	DB2 OLAP Integration Server
Use the model to define a metaoutline on which your cube will be based.	DB2 OLAP Integration Server
Create a command script that will be used to load the cube with data.	DB2 OLAP Integration Server
Create a batch file that runs the command script.	DB2 OLAP Integration Server
Export metadata that defines the batch file to be run from the Data Warehouse Center. Objects that make it possible to schedule the loading and testing of the cube from the Data Warehouse Center are generated by the fignxolv command.	Data Warehouse Center

Creating a star schema from within the Data Warehouse Center

Table 38. Tasks and the product or component you use to perform each task (continued)

Task	Product or component to use
Schedule the step that was created by the export process so that you can populate the cube.	Data Warehouse Center

Designing the warehouse schema in the Data Warehouse Center

Use the Warehouse Schema Model window to create and store warehouse schema models that are associated with a warehouse. The warehouse schema models can be easily exported as metadata to the DB2 OLAP Integration Server (as an OLAP model).

Procedure:

To design a warehouse schema:

1. Define the warehouse schema.
2. Add tables and views to the warehouse schema.
3. Automatically join the tables.
4. Add join relationships between non-keyed columns.

Warehouse schemas

Before you define the warehouse schema you must define the warehouse target tables that you will use as source tables for your warehouse schema:

- When you define a target table that will be used for your warehouse schema, select the **Part of an OLAP schema** check box (in the Define Warehouse Target Table notebook) for the target table that you plan to use as a dimension table or a fact table.
- When you define the warehouse target for a warehouse schema, the warehouse target name must match the name of the physical database on which the warehouse target is defined.

Any warehouse user can define a table in a schema, but only warehouse users who belong to a warehouse group with access to the warehouse target that contains the tables can change the tables.

Only the warehouse schemas that are made up of tables from one database can be exported to the DB2[®] OLAP Integration Server.

Adding tables and views to the warehouse schema

Use the Add Data window to add warehouse target tables, source tables, or source views to the selected warehouse schema. You do not define the tables for the accounts and time dimensions until you export the warehouse schema to the DB2 OLAP Integration Server.

Procedure:

To add the dimension tables and fact table to the warehouse schema:

1. Open the Add Data window:

Creating a star schema from within the Data Warehouse Center

- a. Expand the object tree until you find the **Warehouse Schemas** folder.
 - b. Right-click the warehouse schema, and click **Open**. The Warehouse Schema Modeler window opens.
 - c. Click the **Add Data** icon in the palette, and then click the icon on the canvas at the spot where you want to place the tables. The Add Data window opens.
2. Expand the Warehouse Targets tree until you see a list of tables under the **Tables** folder.
 3. To add tables, select the tables that you want to include in the warehouse schema from the **Available Tables** list, and click > All tables in the **Selected Tables** list contain table icons on the Warehouse Schema Modeler canvas. Click >> to move all the tables to the **Selected Tables** list. To remove tables from the **Selected Tables**, click <. To remove all tables from the **Selected Tables**, click <<.
 4. To create new source and target tables, right-click the **Table** folder in the **Available Tables** tree, and click **Define**. The Define Warehouse Target Table or the Define Warehouse Source Table window opens.
 5. Click **OK**. The tables that you selected are displayed in the window.

Related concepts:

- “Warehouse schemas” on page 202

Related tasks:

- “Designing the warehouse schema in the Data Warehouse Center” on page 202

Exporting a warehouse schema to the DB2 OLAP Integration Server

Use the Export Metadata - OLAP Integration Server window to export warehouse schemas to the OLAP Integration Server (as OLAP models).

Procedure:

To export a warehouse schema to the DB2 OLAP Integration Server:

1. To open the Export Metadata - OLAP Integration Server notebook, from the Data Warehouse Center, right-click **Warehouse**, and select **Export —> OLAP Integration Server** to open the Export Metadata - OLAP Integration Server notebook.
2. Specify the information to export the schema.

You can view the log file that stores trace information about the export process. This file is located in the directory specified by the VWS_LOGGING environment variable. The default value of the VWS_LOGGING variable for Windows NT is x:\sql11b\logging, where x is the drive where the DB2 Universal Database is installed. The name of the log file is FLGNXHIS.LOG.

Related tasks:

- “Exporting warehouse schemas to the OLAP Integration Server: Data Warehouse Center help”

Working with a warehouse schema in the DB2 OLAP Integration Server

This section describes how to work with the warehouse schema in the DB2 OLAP Integration Server.

Completing the design for your cube

After you export the warehouse schema that you designed in the Data Warehouse Center, use the DB2 OLAP Integration Server to complete the design for your cube.

Procedure:

To complete the design of your cube:

1. Optional: View the warehouse schema that you exported by opening the OLAP model (warehouse schema) with the warehouse schema name that you used in the Data Warehouse Center. Make sure that you specify the warehouse target that you used to define the warehouse schema as the data source for the model.
2. Create an OLAP metaoutline, which is a subset of the OLAP model (warehouse schema) on which you will base your cube. Within the metaoutline, you define the hierarchies among the dimensions of the OLAP model. For detailed information about creating a metaoutline based on an OLAP model, see the *OLAP Integration Server Metaoutline User's Guide* for Version 7.1 of the DB2 OLAP Integration Server. For Version 8.1, this information is in the OLAP Model Tutorial, which is available from the DB2 OLAP Integration Server desktop.
3. Create an outline that describes all the elements that are necessary for the Essbase database where your cube is defined. For example, your outline will contain the definitions of members and dimensions, members, and formulas.
4. Define the script that is used to load the cube with data.
5. Define a batch file from which to invoke the script.
6. Export the metadata that defines the batch file to the Data Warehouse Center so that you can schedule the loading of the cube on a regular basis.

Related tasks:

- "Creating an outline in the DB2 OLAP Integration Server" on page 204
- "Creating a load script to load data into the cube in the DB2 OLAP Integration Server" on page 205
- "Creating a batch file to load the command script for the DB2 OLAP Integration Server" on page 206
- "Exporting the metadata to the Data Warehouse Center" on page 207

Creating an outline in the DB2 OLAP Integration Server

This topic describes how to create an outline. After you create the outline, you can associate it with a script that loads data into a cube. The instructions in this topic are based on DB2 OLAP Integration Server Version 7.1; for the latest instructions for using DB2 OLAP Integration Server, see the documentation for the version you are using.

See the online help for the DB2 OLAP Integration Server for detailed information on fields and controls in a window.

Creating a star schema from within the Data Warehouse Center

Procedure:

To create a database outline from the DB2 OLAP Integration Server desktop:

1. Open the metaoutline that you created based on the OLAP model (warehouse schema).
2. Click **Outline** —> **Member and Data Load**. The Essbase Application and Database window opens.
3. In the **Application Name** field, select the name of the OLAP application that will contain the Essbase database into which you want to load data. You can also type a name.
4. In the **Database Name** field, type the name of the OLAP database into which you want to load data.
5. Type any other options in the remaining fields, and click **Next**.
6. Type any other options in the Command Scripts window, and click **Next**.
7. Click **Now** in the Schedule Essbase Load window.
8. Click **Finish**.

The OLAP outline is created. Next, you must create the load script.

Related tasks:

- “Completing the design for your cube” on page 204
- “Creating a load script to load data into the cube in the DB2 OLAP Integration Server” on page 205

Creating a load script to load data into the cube in the DB2 OLAP Integration Server

After you create an outline, you must create a load script that loads data into the cube. After the outline is loaded with data, the resulting cube can be accessed through a spreadsheet program (such as Lotus 1-2-3 or Microsoft Excel), so you can analyze the data. The instructions in this section are based on DB2 OLAP Integration Server Version 7.1; for the latest instructions on using DB2 OLAP Integration Server, see the documentation for the version you are using.

Procedure:

To create the load script:

1. Open the metaoutline for the warehouse schema.
2. Click **Outline** —> **Member and Data Load**. The Essbase Application and Database window opens.
3. In the **Application Name** field, select the name of the OLAP application that will contain the database into which you want to load data. You can also type a name.
4. In the **Database Name** field, type the name of the OLAP database into which you want to load data.
5. Type any other options in the remaining fields and click **Next**.
6. Type any other options in the Command Scripts window and click **Next**.
7. Click **Only Save Load Script** in the Schedule Essbase Load window.
8. Click **Save Scripts**. The Save Script window opens.
9. Type a name and file extension for the command script file.

Creating a star schema from within the Data Warehouse Center

10. Click **Finish**.

The new command script that loads the multidimensional cube with data is created in the \IS\Batch\ directory. The command script contains the following items:

- The name of the DB2 database that contains the source data for the cube
- The Essbase database that will store the cube
- The OLAP catalog name that will be used for the cube
- The instructions that load the data into the cube
- Any calculation options that you specified when you defined the script

The following example is of a command script named `my_script.script`. The line break on the `LOADALL` entry is not significant. You can type the entry all on one line.

```
LOGIN oisserv
SETSOURCE "DSN=tbc;UID=user;PWD=passwd;"
SETTARGET "DSN=essserv;UID=user;PWD=passwd"
SETCATALOG "DSN=TBC_MD;UID=user;PWD=passwd;"
LOADALL "APP=app1;DBN=db1;OTL=TBC Metaoutline;FLT_ID=1;OTL_CLEAR=N;
CALC_SCRIPT=#DEFAULT#;"
STATUS
```

After you create the outline and command script, you must create a batch file that runs the script. The batch file is used as a parameter for the Data Warehouse Center step that runs the script to load the cube.

Related tasks:

- “Completing the design for your cube” on page 204
- “Creating an outline in the DB2 OLAP Integration Server” on page 204
- “Creating a batch file to load the command script for the DB2 OLAP Integration Server” on page 206
- “Exporting the metadata to the Data Warehouse Center” on page 207

Creating a batch file to load the command script for the DB2 OLAP Integration Server

After you create the load command script, you must create a batch file that runs the script.

Procedure:

To create the batch file, use a text editor and enter commands to invoke the script. You might create a file similar to the one in the following example to run `my_script.script`. Do not enter the line break in this example.

```
"C:\IS\bin\olapicmd" < "C:\IS\Batch\my_script.script" >
"C:\IS\Batch\my_script.log"
```

The `my_script.log` log file shows information about the metadata that is exported to the Data Warehouse Center. It also shows if the export process was successful.

Related tasks:

- “Completing the design for your cube” on page 204
- “Creating an outline in the DB2 OLAP Integration Server” on page 204

Creating a star schema from within the Data Warehouse Center

- “Creating a load script to load data into the cube in the DB2 OLAP Integration Server” on page 205
- “Exporting the metadata to the Data Warehouse Center” on page 207

Exporting the metadata to the Data Warehouse Center

Use the `flgnxolv` command to export the metadata for the batch file (that loads the multidimensional cube) to the Data Warehouse Center. The export process creates objects in the Data Warehouse Center that make it possible load and test the cube.

Prerequisites:

Before you export the metadata, make certain you have already defined the tables for your warehouse schema.

Procedure:

To export metadata to the Data Warehouse Center, enter the following command:

```
flgnxolv HIScatalog HISid HISpw  
HISpre OLAPscript OLAPcube DWcdb  
DWCid DWcpw DWcpre ModelName
```

HIScatalog

The name of the DB2 OLAP Integration Server metadata catalog.

HISid

The supervisor user ID for DB2 OLAP Integration Server.

HISpw

The supervisor user password for DB2 OLAP Integration Server.

HISpre

The table prefix of the DB2 OLAP Integration Server metadata catalog.

OLAPscript

The path and file name of the batch file that invokes the OLAP load script.

OLAPcube

The OLAP cube, which is identified by four objects in the following format:

```
essbaseServer.application.essbaseDatabase.outline
```

essbaseServer

The name of the OLAP server.

application

The name of the OLAP application that contains the database that is identified by *essbaseDatabase*

EssbaseDatabase

The name of the OLAP server database that contains the outline that is identified by *outline*.

outline The name of the OLAP server outline whose metadata you want.

DWCdb

The Data Warehouse Center control database.

DWCid

The Data Warehouse Center control database user ID.

DWCpw

The Data Warehouse Center control database password.

Creating a star schema from within the Data Warehouse Center

DWCpre

The Data Warehouse Center table prefix (IWH).

ModelName

The name of the OLAP model.

The metadata for the batch file is exported to the Data Warehouse Center. See the log file for information about the metadata.

Loading the multidimensional cube from the Data Warehouse Center

This section describes loading the multidimensional cube from the Data Warehouse Center.

Data Warehouse Center objects created by the DB2 OLAP Integration Server export process

When you export metadata from the DB2[®] OLAP Integration Server, the following Data Warehouse Center objects are created and associated with the target tables in the warehouse schema:

- A subject area named "OLAP cubes"
- A process within the subject area named in the following format:

`servername.applicationname.databasename.outlinename`

servername

The name of the OLAP server.

applicationname

The name of the OLAP server application that contains the database that is identified by *databasename*.

databasename

The name of the OLAP server database that contains the outline that is identified by *outlinename*.

outlinename

The name of the OLAP server outline whose metadata you exported.

- A step with a name in the same format as the process.
The step uses the batch file (whose metadata you exported) as a parameter. When you click the **Parameters** tab in the step's Properties notebook, the **Parameter value** column displays the fully qualified name of the batch program that calls the command script that you created in the DB2 OLAP Integration Server. For example, the Parameter value column, might show
`c:\is\batch\my_script.bat`.

When you run the step, the batch file runs the script to load the multidimensional cube.

When you select the process, the tables that comprise the warehouse schema are displayed in the right pane of the Data Warehouse Center. When the step runs, the warehouse schema tables are used as source tables to build and populate the multidimensional cube. The dimension tables are used as sources for the members of the OLAP model, and the fact table is the source for the measures (the data in the cube).

Related concepts:

- "About publishing metadata about your warehouse schema" on page 209

Related tasks:

- “Creating a schedule to populate a cube” on page 209

Creating a schedule to populate a cube

You can schedule the step that loads the multidimensional cube, and you can promote it so that it runs on a regular basis.

Procedure:

To schedule and promote the step:

1. Right-click the step, and click **Schedule**. The Schedule notebook opens.
2. Use the Schedule page to define the intervals at which you want the step to run.
3. Use the Task Flow page of the Schedule notebook to specify that this step should run only upon successful completion of the step that you defined to create the warehouse target tables for the warehouse schema.
4. Right-click the step, and click **Mode**. You can click either **Test** or **Production** to run the step.

After you successfully schedule the step, the cube that was built using your warehouse schema is populated.

Related concepts:

- “Data Warehouse Center objects created by the DB2 OLAP Integration Server export process” on page 208
- “About publishing metadata about your warehouse schema” on page 209

About publishing metadata about your warehouse schema

You can use the Publish Metadata - Data Warehouse to Information Catalog notebook to publish to the information catalog the metadata that describes the tables in your warehouse schema. A warehouse schema maps to a star schema in the Information Catalog Center.

Related tasks:

- “Publishing warehouse metadata” on page 174
- “Preparing to publish OLAP server metadata” in the *Information Catalog Center Administration Guide*

Related reference:

- “Metadata mappings between the Information Catalog Center and OLAP server” in the *Information Catalog Center Administration Guide*

Creating a star schema from within the Data Warehouse Center

Chapter 17. Maintaining the warehouse database

This chapter explains how to maintain your warehouse database using the following warehouse utilities:

- DB2 UDB REORG
- DB2 for z/OS REORG
- DB2 UDB RUNSTATS
- DB2 z/OS utilities

Reorganizing data

You can reorganize data in a DB2 Universal Database table or in a DB2 Universal Database for z/OS table space or index using the DB2 UDB REORG or DB2 for z/OS REORG utilities.

Defining values for DB2 UDB REORG or DB2 for z/OS REORG utilities

You can use the DB2 reorganize utilities to rearrange a table in physical storage. Rearranging a table in physical storage eliminates fragmentation and ensures that the table is stored efficiently in the database. You can also use reorganization to control the order in which the rows of a table are stored, usually according to an index.

Procedure:

To define values for DB2 Universal Database REORG or the DB2 for z/OS REORG utilities, open the Properties notebook for the step that you want to define and specify the necessary values.

Defining values for a DB2 for z/OS Utility

Use the DB2 for z/OS Utility to run any utility supported by DSNUTILS.

Procedure:

To define values for a step that uses this utility:

1. Add a target table to the process.
2. Add a DB2 reorg step to the process.
3. Link the target table to the step.
4. Open the Properties notebook for the step, and specify the step's properties.
5. Click **OK** to save your changes and close the notebook.

Parameters for a DB2 for z/OS Utility

Here are the parameters for a DB2 for z/OS Utility and their definitions.

UTILITY_ID

Specifies a unique identifier (UID) for this utility within DB2. This is a string of 1-16 characters.

Maintaining the warehouse database

RESTART

Specifies whether this utility restarts a current utility, and, if so, at what point it is to be restarted. NO or null indicates the utility is new, not a restart. There must not be any other utility with the same utility identifier (UID). The default is null.

UTSTMT

Specifies the utility control statements. This is an input parameter of type VARCHAR(32704).

RETCODE

Specifies the utility highest return code. This is an output parameter of type INTEGER.

UTILITY_NAME

Specifies the utility that you want to start. Specify only a single utility that requires data set allocation in the UTSTMT parameter.

RECDSN

Specifies the cataloged data set name that is required when reorganizing table spaces as the unload data set. If you specify RECDSN, it will be allocated to the SYSREC DDNAME.

RECDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set listed in RECDSN resides.

RECSpace

Specifies the number of cylinders to use as the primary space allocation for the data set specified by RECDSN. The secondary space allocation will be 10% of the primary space allocation.

DISCDSN

Specifies the cataloged data set name used when reorganizing table spaces as a discard data set to hold records not reloaded. If you specify DISCDSN, it will be allocated to the SYSDISC DDNAME.

DISCDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by DISCDSN resides.

DISCSpace

Specifies the number of cylinders to use as the primary space allocation for the data set specified by DISCDSN. The secondary space allocation will be 10% of the primary space allocation.

PNCHDSN

Specifies the cataloged data set name that is used when reorganizing table spaces with the keywords UNLOAD EXTERNAL or DISCARD. The data set is used to hold the generated LOAD utility control statements. If you specify a value for PNCHDSN, it will be allocated to the SYSPUNCH DDNAME.

PNCHDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by PNCHDSN resides.

PNCHSpace

Specifies the number of cylinders to use as the primary space allocation for the data set specified by PNCHDSN. The secondary space allocation will be 10% of the primary space allocation.

COPYDSN1

Specifies the name of the target (output) data set. If you specify COPYDSN1, it will be allocated to the SYSCOPY DDNAME.

COPYDEVT1

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by COPYDSN1 resides.

COPYSPACE1

Specifies the number of cylinders to use as the primary space allocation for the data set specified by COPYDSN1. The secondary space allocation will be 10% of the primary space allocation.

COPYDSN2

Specifies the name of the cataloged data set used as a target (output) data set for the backup copy. If you specify COPYDSN2, it will be allocated to the SYSCOPY2 DDNAME.

COPYDEVT2

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by COPYDSN2 resides.

COPYSPACE2

Specifies the number of cylinders to use as the primary space allocation for the data set specified by COPYDSN2. The secondary space allocation will be 10% of the primary space allocation.

RCPYDSN1

Specifies the name of the cataloged data set used as a target (output) data set for the remote site primary copy. If you specified RCPYDSN1, it will be allocated to the SYSRCPY1 DDNAME.

RCPYDEVT1

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the RCPYDSN1 data set resides.

RCPYSPACE1

Specifies the number of cylinders to use as the primary space allocation for the data set specified by RCPYDSN1. The secondary space allocation will be 10% of the primary space allocation.

RCPYDSN2

Specifies the name of the cataloged data set used as a target (output) data set for the remote site backup copy. If you specify RCPYDSN2, it will be allocated to the SYSRCPY2 DDNAME.

RCPYDEVT2

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by RCPYDSN2 resides.

RCPYSPACE2

Specifies the number of cylinders to use as the primary space allocation for the data set specified by RCPYDSN2. The secondary space allocation will be 10% of the primary space allocation.

WORKDSN1

Specifies the name of the cataloged data set that is required as a work data set for sort input and output. If you specify WORKDSN1, it will be allocated to the SYSUT1 DDNAME.

Maintaining the warehouse database

WORKDEVT1

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by WORKDSN1 resides.

WORKSPACE1

Specifies the number of cylinders to use as the primary space allocation for the data set specified by WORKDSN1. The secondary space allocation will be 10% of the primary space allocation.

WORKDSN2

Specifies the name of the cataloged data set that is required as a work data set for sort input and output. It is required if you are using reorganizing non-unique type 1 indexes. If you specify WORKDSN2, it will be allocated to the SORTOUT DDNAME.

WORKDEVT2

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by WORKDSN2 resides.

WORKSPACE2

Specifies the number of cylinders to use as the primary space allocation for the WORKDSN2 data set. The secondary space allocation will be 10% of the primary space allocation.

MAPDSN

Specifies the name of the cataloged data set that is required as a work data set for error processing during LOAD with ENFORCE CONSTRAINTS. It is optional for LOAD. If you specify MAPDSN, it will be allocated to the SYSMAP DDNAME.

MAPDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by MAPDSN resides.

MAPSPACE

Specifies the number of cylinders to use as the primary space allocation for the MAPDSN data set. The secondary space allocation will be 10% of the primary space allocation.

ERRDSN

Specifies the name of the cataloged data set that is required as a work data set for error processing. If you specify ERRDSN, it will be allocated to the SYSERR DDNAME.

ERRDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by ERRDSN resides.

ERRSPACE

Specifies the number of cylinders to use as the primary space allocation for the ERRDSN data set. The secondary space allocation will be 10% of the primary space allocation.

FILTRDSN

Specifies the name of the cataloged data set that is required as a work data set for error processing. If you specify FILTRDSN, it will be allocated to the FILTER DDNAME.

FILTRDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by FILTRDSN resides.

FILTRSPACE

Specifies the number of cylinders to use as the primary space allocation for the FILTRDSN data set. The secondary space allocation will be 10% of the primary space allocation.

Related tasks:

- “Creating views for MQSeries messages” on page 176

Updating system catalog statistics

Use the DB2 UDB RUNSTATS or DB2 for z/OS utilities to update system catalog statistics.

Defining values for a DB2 UDB RUNSTATS utility

Use the Properties notebook for the DB2 UDB RUNSTATS utility to create a step that can be used to update system catalog statistics on the data in a table, the data in the table indexes, or the data in both the table and its indexes. The optimizer uses these statistics to choose which path will be used to access the data.

In general, you need to update statistics if there are extensive changes to the data in the table.

Procedure:

To define values for a step that uses this utility:

1. Add a target table to the process.
2. Add a DB2 UDB RUNSTATS step to the process.
3. Link the target table to the step.
4. Open the Properties notebook for the step, and specify the step’s properties.
5. Click **OK** to save your changes and close the notebook.

Related tasks:

- “Running statistics on a table: Data Warehouse Center help”

Defining values for a DB2 for z/OS RUNSTATS utility

You can use the DB2 z/OS RUNSTATS utility to gather summary information about the characteristics of data in table spaces.

Procedure:

To define values for a DB2 z/OS RUNSTATS utility, open the step notebook and specify information about the warehouse program, parameters, and processing options.

Chapter 18. Managing the control database

This chapter explains how to manage the warehouse control database.

Backing up data

This section explains how to back up the data in your warehouse database.

Stopping the Data Warehouse Center services (Windows)

Before you back up the warehouse database, you must stop the Data Warehouse Center services.

Procedure:

To stop the Data Warehouse Center services:

1. On the workstation that contains the warehouse server, click **Settings** → **Control Panel** → **Services**. The Services window opens.
2. Select a warehouse service, and click **Stop**.
3. Repeat this step for each warehouse service that is listed in the Services window.

If you stop the warehouse logger, the server also stops. However, if you stop the warehouse server, the logger does not stop.

4. When all warehouse services are stopped, click **Close**.

Backing up the warehouse control database

Back up your warehouse control database, as well as all your warehouse databases, at regular intervals.

Prerequisites:

Stop the Data Warehouse Center services before you back up the warehouse control database.

Procedure:

To back up the warehouse control database, use the standard procedures for DB2 backup and recovery.

Related tasks:

- “Stopping the Data Warehouse Center services (Windows)” on page 217

Expanding your warehouse

With the Data Warehouse Center, you can expand your network configuration as your warehouse grows. You can:

- Add new workstations with new warehouse agents and administrative interfaces.
- Delete agents and administrative interfaces from your existing configuration.

Managing the control database

- Create a warehouse control database. If you create a warehouse control database, you must run the initialization process again.

Adding or deleting administrative interfaces and warehouse agents in the Data Warehouse Center

You can alter the configuration of your Data Warehouse Center by adding or deleting administrative interfaces and warehouse agents.

Procedure:

To add administrative interfaces or warehouse agents to your Data Warehouse Center configuration, follow the installation process for those components.

To delete administrative interfaces or warehouse agents from your Data Warehouse Center configuration, remove those components.

Related concepts:

- “Agent site configurations” on page 17

Related tasks:

- “Defining an agent site” on page 18

Initializing a warehouse database

This section explains how to change the active control database and how to initialize a control database during installation.

Changing the active warehouse control database

When you install the warehouse server, the warehouse control database that you specify during installation is initialized. During a typical installation, a default control database called DWCTRLDB is created and initialized. *Initialization* is the process in which the Data Warehouse Center creates the control tables that are required to store Data Warehouse Center metadata. To determine the name of the active control database, click **Advanced** in the Data Warehouse Center Logon window.

To use a control database other than the active control database, use the Warehouse Control Database Management tool to switch databases. The Warehouse Control Database Management tool registers the database that you want to use as the active warehouse control database. Before using the Warehouse Control Database Management tool, you must stop the warehouse server.

The Data Warehouse Center creates the database that you specify on the warehouse server workstation if the database does not already exist on the workstation. If you want to use a remote database, create the database on the remote system and catalog it on the warehouse server workstation.

The DB2 Control Center or the DB2 Command Line Processor might indicate that the warehouse control database is in an inconsistent state. This message is expected because the warehouse server did not commit its initial startup message to the warehouse logger.

Procedure:

To initialize a new warehouse control database:

1. Stop the warehouse server.
 - On Windows, stop the warehouse server.
 - On AIX, stop the warehouse server and logger by entering the command `db2vsvr stop`.
2. Open the Warehouse Control Database Management window.
 - On Windows, click **Start** → **Programs** → **IBM DB2** → **Set-up Tools** → **Warehouse Control Database Management**.
 - On AIX, run the script `db2wcdbm`.
3. In the **New control database** field, type the name of the database that you want to use as your warehouse control database.
4. In the **Schema** field, type the name of the schema to use for the database. Schema name is case sensitive.
5. In the **User ID** field, type the user ID that has access to the database.
6. In the **Password** field, type the password for the user ID.
7. In the **Verify Password** field, type the password again.
8. Click **OK**.

The window remains open. The **Messages** field displays messages that indicate the status of the creation and migration process.
9. After the process is complete, click **Cancel** to close the window.
10. Start the Data Warehouse Center administrative interface.
11. Click **Advanced** in the Data Warehouse Center Logon window, and type the name of the new warehouse control database in the **Control database** field.
12. Click **OK** to save your changes.

Related tasks:

- “Starting and stopping the warehouse server and logger (Windows)” on page 9
- “Initializing a warehouse control database during installation” on page 219

Initializing a warehouse control database during installation

To initialize a warehouse control database other than the default control database that is created during installation, you must specify the name of the control database that you want to use during installation. The installation process will create the database with the name that you specify, initialize the database for use with the Data Warehouse Center, and register the database as the active warehouse control database.

If you apply a fix pack or install a new release of DB2 or the Data Warehouse Center, you must migrate the existing control database to update the objects that it contains. In addition, after you install Fix Pack 2, you must re-initialize the warehouse control database.

Procedure:

To create a warehouse control database: during installation on a workstation other than where the warehouse server is installed:

Managing the control database

1. During installation on a workstation other than where the warehouse server is installed, select **Warehouse Local Control Database** during a custom installation. The installation process will create the database.
2. After installation, you must use the Warehouse Control Database Management window on the warehouse server workstation by following the steps for initializing a new warehouse control database.
3. Specify the database name that you specified during installation. The database will be initialized for use with the Data Warehouse Center and registered as the active warehouse control database.

Migrating a DB2 Universal Database Version 7 Control Database to DB2 Universal Database Version 8 when Version 7 and Version 8 are installed on different systems

If your DB2 Universal Database, Version 7 warehouse is installed on a different system than the system on which you installed DB2 Universal Database, Version 8, follow these instructions to migrate your Version 7 warehouse control database to Version 8.

Procedure:

To migrate a DB2 Universal Database, Version 7 warehouse control database to DB2 Universal Database, Version 8 when Version 7 and Version 8 are installed on different systems:

1. On the DB2 Universal Database, Version 7 system:
 - a. Create a directory to contain the files produced by the db2move operation, and then change to this directory.
 - b. Enter the following command on your operating system's command line or the DB2 command line:

```
db2move v7controldatabasename export
```

where *v7controldatabasename* is the name of the Version 7 warehouse control database that you want to migrate.
2. On the DB2 Universal Database, Version 8 system:
 - a. Move the directory that was produced as a result of the db2move operation to the computer where DB2 Universal Database, Version 8 is installed, and use the operating system command line window to change to this directory.
 - b. Create a DB2 Universal Database, Version 8.1 database by using the DB2 Control Center or the DB2 Command Line Processor.
 - c. In the operating system command line window, enter the following command:

```
db2move v8controldatabasename import
```

where *v8controldatabasename* is the name of the Version 8 database you just created.
 - d. Use the Warehouse Control Database Management window to initialize the Version 8 warehouse control database.

Related tasks:

- "Backing up the warehouse control database" on page 217
- "Changing the active warehouse control database" on page 218
- "Initializing a warehouse control database during installation" on page 219

Related reference:

- “db2move - Database Movement Tool Command” in the *Command Reference*

Data Warehouse Center configuration

This section explains how to configure the Data Warehouse Center.

Data Warehouse Center configuration

You can use the Data Warehouse Center Properties notebook to change global settings for your Data Warehouse Center installation. You can override many global settings in the objects that use them. For example, you can use the Properties notebook to specify the default behavior of a processing step when the warehouse agent finds no rows in the source table or file. You can override this global setting in a particular step.

You can use the configuration tool only if the Data Warehouse Center server is installed on the workstation (as well as the administrative client).

Important: Some of the fields in the Properties notebook should be changed only on the recommendation of IBM® Software Support. For more information, see the online help for the Data Warehouse Center Properties notebook.

Related tasks:

- “Adding or deleting administrative interfaces and warehouse agents in the Data Warehouse Center” on page 218
- “Initializing a warehouse control database during installation” on page 219

Managing the control database

Chapter 19. Refreshing an OLAP Server database

You can use warehouse programs to automatically refresh a OLAP Server database. The Data Warehouse Center can help you schedule three types of operations with an OLAP Server:

- Loading data
- Updating dimensions
- Running calculations

Loading data into the OLAP server database from the Data Warehouse Center

You can use the Data Warehouse Center to load data into the OLAP server database.

Prerequisites:

To use the Data Warehouse Center OLAP server programs, you must install and configure the IBM DB2 OLAP Server software. You can install this software locally on an agent site, or on another system on the network.

Procedure:

To use the Data Warehouse Center to load data into the OLAP server database:

1. Using the Essbase Application Manager, create the OLAP Server application and database. Make note of the application name, the database name, the user ID, and the password. You will need this information as input to a warehouse program.
2. Using the Essbase Application Manager, define the outline for the database.
3. Define the data that you want to extract from the operational sources for OLAP Server to load into the Essbase database. You can use this data to update measures (for example, using the Essbase IMPORT command) and dimensions (for example, using the BuildDimension command).
4. Define a step that extracts data from the operational data sources and builds the data as defined in step 3.
5. Promote the step to test mode and run it at least once.
6. Using the Essbase Application Manager, write and test the load rules that will load the data sources into the Essbase database. Save the load rules into the database or as files on the warehouse agent site.
You can also define calculation scripts to run after the data is loaded. Save the calculation scripts in files on the warehouse agent site.
7. Define a step that uses one of the Data Warehouse Center OLAP server programs, such as Load data from flat file with load rules (ESSDATA2). Use the Process Model window to specify that the step that extracts data is to start this step.
8. Promote the step to test mode and run it at least once.
9. Define a schedule for the step that extracts data, and promote the step to production mode.

Refreshing an OLAP Server database

The following picture shows the data flow between the Data Warehouse Center and OLAP server.

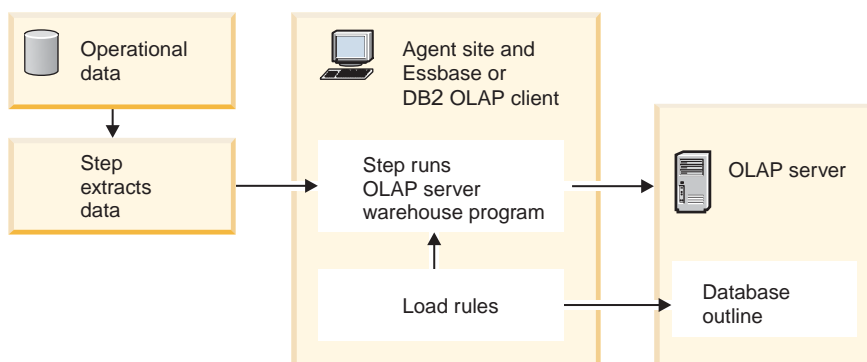


Figure 17. Data flow between the Data Warehouse Center and OLAP server

Running calculations on OLAP server data from the Data Warehouse Center

You can use the following programs to run calculations on OLAP server data from the Data Warehouse Center:

- Default calc (ESSCALC1) warehouse program
- Calc with calc rules (ESSCALC2)

Procedure:

To call the default calc script that is associated with the target database, add a Default calc (ESSCALC1) warehouse program to a process, and define the parameters for the program.

To apply a calc script to an OLAP server database from the Data Warehouse Center, add a Calc with calc rules (ESSCALC2) warehouse program to a process, and define the parameters for the program.

Loading data from a flat file into an OLAP Server database

You can load data from a flat file into an OLAP server database from the Data Warehouse Center.

Procedure:

To load data from a flat file into an OLAP server database, add a Free text data load (ESSDATA1) warehouse program to a process, and define the parameters for the load process.

To load data from a flat file into an OLAP server database using load rules, add a Load data from file with load rules (ESSDATA2) warehouse program to a process, and define the parameters for the load process.

Loading data from a table into an OLAP server database

You can load data from a table into an OLAP server database from the Data Warehouse Center.

Procedure:

To load data from an SQL table into an OLAP server database using load rules, add a Load data from SQL table with load rules (ESSDATA3) warehouse program to a process, and define the parameters for the load process

To load data from a table into an OLAP server database without using load rules, add a Load data from a file without using load rules (ESSDATA4) warehouse program to a process, and define the parameters for the load process.

Updating an OLAP server outline from the Data Warehouse Center

Use the OLAP server programs in the Data Warehouse Center to update an OLAP server outline.

Procedure:

To update an OLAP server outline from a source flat file using load rules, add a Update Outline from file (ESSOTL1) warehouse program to a process, and define parameters for the load process.

To update an OLAP server outline from an SQL source using load rules, add an Update outline from SQL table (ESSOTL2) warehouse program to a process, and define parameters for the load process.

Chapter 20. Data Warehouse Center logging and trace data

The Data Warehouse Center provides three levels of logging capability:

- Basic logging function
- Program trace
- Start error trace

To assist you in diagnosing problems, each of these logging levels contains a variety of features. This chapter describes the Data Warehouse Center logging levels and features.

The basic logging function

The basic logging function in the Data Warehouse Center captures all the events and errors that occur during run time and build time. This information is stored in the warehouse control database. The log viewer displays these run-time and build-time events and errors and any messages or return codes that pertain to them.

Warehouse log files

The log holds records until a designated count limit is reached. When the count limit is reached, the Data Warehouse Center automatically deletes the oldest logs.

Recommendation: Set the log record count to a size that holds 3 to 4 days worth of records.

You cannot turn off the basic logging function.

Related tasks:

- “Viewing build-time errors using the basic logging function” on page 228
- “Viewing log entries in the Data Warehouse Center” on page 228

Viewing run-time errors using the basic logging function

Use the basic logging function to view run-time (step processing) errors.

Procedure:

To view run-time errors:

1. Open the Data Warehouse Center desktop.
2. Click **Data Warehouse Center** → **Work in Progress**.
The Work in Progress window opens.
3. Select the step for which you want to view errors.
4. To open the Log Viewer window and display the run-time errors for the selected step, click **Log**.

Viewing build-time errors using the basic logging function

Use the basic logging function in the Data Warehouse Center to view build-time errors.

Procedure:

To view build-time (table import, object creation, and step promotion) errors:

1. Open the Work in Progress window.
2. Click **Work in Progress** → **Show Log** to open the Log Viewer window and display the build-time errors for the Data Warehouse Center.

Viewing log entries in the Data Warehouse Center

If a step or process does not run successfully, you can use the Log Viewer to find the cause of the failure.

Procedure:

To view log entries:

1. Select the step or process.
2. Click **Log** to open the Log Viewer window.

Data Warehouse Center component trace data

This section describes component trace data and how to run a Data Warehouse Center Component trace.

Component trace data

Run a Data Warehouse Center trace at the direction of IBM® Software Support to produce a record of a program execution. You can run an ODBC trace, a trace on the warehouse control database, and traces on the warehouse server, agent, and logger components.

When you run a trace, the Data Warehouse Center writes information to text files. Data Warehouse Center programs that are called from steps will also write any trace information to this directory. These files are located in the directory specified by the *VWS_LOGGING* environment variable.

For the iSeries™ system, many Warehouse Center trace files are stored in the iSeries Integrated File System. To edit these trace files, you can either use FTP to move these files to the workstation or use Client Access for iSeries.

Recommendation: Periodically delete the trace files from the program files\sqlib\logging directory, so that they will not contain obsolete data.

The default value of *VWS_LOGGING* is:

Windows® NT, Windows 2000, Windows XP
x:\program files\sqlib\logging

UNIX® and z/OS™
/var/IWH

iSeries

/QIBM/UserData/IWH

The Data Warehouse Center writes these files on Windows NT®, Windows 2000, and Windows XP:

AGNT $nnnn$.LOG

Contains trace information. Where $nnnn$ is the numeric process ID of the warehouse agent, which can be 4 or 5 characters depending on the operating system.

AGNT $nnnn$.SET

Contains environment settings for the agent. Where $nnnn$ is the numeric process ID of the warehouse agent, which can be 4 or 5 characters depending on the operating system.

IWH2LOG.LOG

Contains the results of the trace for the logger component.

IWH2SERV.LOG

Contains the results of the trace for the warehouse server.

IWH2DDD.LOG

Contains the results of the trace for the warehouse control database.

IWH2RG nn .LOG

Contains the results of migration commands.

If you are running a UNIX agent, the Data Warehouse Center writes the following files on the UNIX workstation:

startup.log

Contains trace information about the startup of the warehouse agent daemon.

vwdaemon.log

Contains trace information about warehouse agent daemon processing.

One common problem is caused by running Windows NT, Windows 2000, or Windows XP warehouse agents as a system process rather than a user process. When the warehouse agent runs as a system process, it is not authorized to connect to network drives or products because the process does not have a user ID. Symptoms of this problem include the warehouse agent being unable to find the warehouse program (**Error RC2 = 128** or **Error RC2 = 1** in the Log Viewer Details window) or being unable to initialize the program.

If the warehouse agent runs as a user process, the warehouse agent has the characteristics of the user, including the ability to access network drives or programs to which the user is authorized.

To avoid these problems, run the warehouse agent as a user process. If you are using the default agent, run the warehouse server as a user process.

If the value of the environment variable, `VWS_SERVER_LOG_MAX` is greater than 0, the warehouse server will stop enlarging the log file when it reaches a size that is approximately equal to the number of bytes indicated by the value of `VWS_SERVER_LOG_MAX`. When the log file reaches the maximum size, the newest trace log entries are retained and the oldest entries are overwritten. When you are doing extensive tracing, `VWS_SERVER_LOG_MAX=150000000` (150M) is a reasonable size.

Related tasks:

- “Running warehouse agents as a user process (Windows)” on page 230
- “Running a Data Warehouse Center component trace” on page 230

Running warehouse agents as a user process (Windows)

If you run a warehouse agent as a system process instead of a user process, it can cause steps to fail. When the warehouse agent runs as a system process, it is not authorized to connect to network drivers or products because the process does not have a user ID. If the warehouse agent runs as a user process, the warehouse agent has the characteristics of a user, including the ability to access network drives or programs to which the user is authorized.

Procedure:

To run the warehouse agent as a user process:

1. Change the warehouse server, warehouse logger, and warehouse agent daemon services to run as user processes by performing the following steps:
 - a. Double-click the **Services** icon in the Windows **Control Panel** folder.
 - b. Stop the services.
 - c. Select the service, and click **Startup**.
 - d. Click **This Account**.
 - e. Click the ... button after the **This Account** field to select a user ID.
The user ID must have administration authority in Windows NT, Windows 2000, or Windows XP and authorization to any required network drive.
 - f. Type the password for the user ID in the appropriate fields.
 - g. Click **OK**.
 - h. Restart the services.
2. If you are using a supplied OLAP server program, verify that the DB2 OLAP or Essbase client is installed on a drive that is local to the agent that is running the program.
3. If you are using a supplied OLAP server program, verify that the ARBORPATH variable (set on the Essbase client or administrator) specifies a drive that is local to the agent that is running the program and is specified as a system variable.

Related tasks:

- “Stopping the Data Warehouse Center services (Windows)” on page 217

Running a Data Warehouse Center component trace

Follow this procedure to run a Data Warehouse Center component trace.

Procedure:

To run a Data Warehouse Center component trace:

1. Right-click the warehouse object, and click **Properties**.
2. Specify the trace level for the warehouse control database, ODBC connection, server, agent, or logger as directed by IBM Software Support.
3. Click **OK**.
4. Restart the services as requested.

5. Perform the failing operation.
6. Repeat steps 1 through 4 to set the trace level back to 0.

After completing this procedure, turn the trace level back to 0 to prevent performance degradation.

You can run an agent trace independently for individual steps by setting the trace level in the step's Properties notebook on the Processing options page.

Error logging for warehouse programs and transformers

The supplied warehouse programs and transformers write errors to log files.

Warehouse programs

Supplied warehouse programs write data to the directory that is specified in the `VWS_LOGGING` environment variable. Clear the directory of the log files after sending the log files to IBM® Software Support.

Transformers

Transformer error messages start with `DWC14`. Transformer error messages, warning messages, and returned SQL codes are stored as secondary codes. For example, if a message starts with `DWC14`, a transformer (stored procedure) caused the error. If the secondary code includes an `SQLCODE`, an SQL statement in the transformer caused the error. To enable logging, specify a log table name on the Processing Options page of the Properties notebook for the step, and add a suffix of `:n` to the log table name. The value of `n` indicates the logging level:

- | | |
|---|---|
| 0 | No logging |
| 1 | Log errors only |
| 2 | Log errors and warnings (this is the default logging level) |
| 3 | Log errors, warnings, and informational messages (for example, starting and stopping a transformer) |

For example, to indicate a log table named `MyLogTable` that contains log entries at log level 3 or less, specify `MyLogTable:3`. In the output log tables, the message type is one of the following values:

- | | |
|---|----------|
| E | Error |
| W | Warning |
| Q | SQL code |

You can include a table space name after the log table name by appending the log level to the table space name.

For example, to indicate a log table named `MyLogTable` that is located in the `MyTableSpace` table space and contains entries at log level 3 or less, specify `MyLogTable,MyTableSpace:3`.

The output log table in the warehouse control database contains detailed error messages, warning messages, and SQL codes. In the output log tables, the message type is one of the following values:

- | | |
|---|---------|
| E | Error |
| W | Warning |

Data Warehouse Center logging and trace data

Q SQL code

Related concepts:

- “Warehouse log files” on page 227

Related tasks:

- “Viewing build-time errors using the basic logging function” on page 228
- “Viewing log entries in the Data Warehouse Center” on page 228

Tracing Apply program errors

You can trace errors that are created when you use the Apply program.

Procedure:

To enable tracing for the Apply program, set the Agent Trace value = 4 in the Warehouse Properties page. The Agent turns on full tracing for Apply when Agent Trace = 4.

If you do not see any data in the CD table, then the Capture program is not started, or you did not create changed data by updating the source table.

Start error trace files

The Data Warehouse Center creates three log files automatically when the logger is not running. The log file names are IWH2LOGC.LOG, IWH2LOG.LOG, and IWH2SERV.LOG. The Data Warehouse Center stores the files in the directory that is specified by the VWS_LOGGING environment variable.

The log files are:

IWH2LOGC.LOG

When the logger is not running, processes write messages to this file. The Data Warehouse Center server and the OLE server write to this file. The file is created only if the logger stops. The file contains the complete content of all messages that could not be sent.

IWH2LOG.LOG

The logger creates this file when it cannot start itself or when a trace is activated. Key diagnostic information is written to this file when the logger cannot start itself, and cannot write to the Data Warehouse Center log. If you hear five beeps, or receive an application error, when the logger stops, look in this file. The server cannot start if the logger cannot start.

IWH2SERV.LOG

The server log contains the startup message and grows when the server trace is on.

Appendix A. Metadata mappings

This appendix covers metadata mappings between the Data Warehouse Center and the following programs:

- DB2 OLAP Integration Server metadata to Data Warehouse Center metadata
- ERwin metadata to Data Warehouse Center metadata
- Trillium Software System to Data Warehouse Center metadata
- CWM XML metadata to Data Warehouse Center metadata

See the *Information Catalog Center Administration Guide* for information about Information Catalog Center to Data Warehouse Center and Information Catalog Center metadata to OLAP server metadata mapping.

Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center

The following table shows the mapping of DB2 OLAP Integration Server metadata to Data Warehouse Center metadata. You use the following metadata mappings when you want to publish data from the OLAP Integration Server to the Information Catalog Center.

Table 39. Metadata mappings between DB2 OLAP Integration Server and the Data Warehouse Center

DB2 OLAP Integration Server metadata	Data Warehouse Center metadata tag language
N/A	SubjectArea – OLAP Cubes
OLAP cube name	Process name
This property value is the 4-part name in the following format: server.application.database.outline	
OLAP cube name	Step name
This property value is the 4-part name in the following format: server.application.database.outline	
Data source	SourceDataBase (pre-existing in the control database)
This property is based on a query of the DB2 OLAP Integration Server catalog (OA_INFO table) based on the following column: PHYSICAL_TABLE.	StepInputTable A StepInputTable object is created for each table in the source database.
This property is based on a query of the DB2 OLAP Integration Server catalog (OM_INFO table) based on the following columns: Columns MODEL_NAME and MODEL_IDget MODEL_DATA_SOURCE	DatabaseName

Metadata mappings

Table 39. Metadata mappings between DB2 OLAP Integration Server and the Data Warehouse Center (continued)

DB2 OLAP Integration Server metadata	Data Warehouse Center metadata tag language
This property is based on a query of the DB2 OLAP Integration Server catalog (OA_INFO table) based on the following columns: MODEL_NAME and MODEL_ID get PHYSICAL_TABLE Only the name on the right side of the period (without the schema) is used.	TablePhysicalName
OLAP cube name	Cube name
This property value is the 4-part name in the following format: server.application.database.outline	
N/A	StepViewVWPOutputTable
OLAP cube name	Step name
This property value is the 4-part name in the following format: server.application.database.outline	
N/A	VWPPProgramTemplate – Build OLAP Cube
N/A	VWPPProgramTemplateParameter
N/A	VWPPProgramInstance
N/A	VWPPProgramInstanceParameter
DB2 OLAP Integration Server bat file name	VWPPProgramInstanceParameterName
N/A	VWP Group – OLAP Integration

Related tasks:

- “Importing metadata” on page 173
- “Metadata export capabilities” on page 171

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Mapping ERwin Version 3.5.2 object attributes to Data Warehouse Center tags” on page 236
- “Metadata mappings between the Data Warehouse Center and CWM XML objects and properties” on page 238
- “Mapping ERwin Version 4.0 and later object attributes to Data Warehouse Center tags” on page 234

Mapping ERwin Version 4.0 and later object attributes to Data Warehouse Center tags

The following tables show how the main ERwin Version 4.0 object attributes in an XML file correspond to the Data Warehouse Center tags:

This table shows the model to WarehouseDatabase.tag or SourceDatabase.tag mapping.

Table 40. The model to WarehouseDatabase.tag or SourceDatabase.tag mapping

XML tag	Data Warehouse Center tag
ModelProps.Name	DatabaseName
ModelProps.Definition	DatabaseDescription
Not applicable	DatabaseNotes
ModelProps.Author	DatabaseContact
Not applicable	DatabaseServerName
ModelProps.Name	DatabasePhysicalName
ModelProps.Target_Server*	DatabaseType
ModelProps.Target_Server*	DatabaseTypeExtended
Not applicable	DatabaseUserid

* The following database types are supported: DB2, Informix, Oracle, Sybase, Microsoft SQL Server, and generic ODBC. The appropriate value is mapped to the DatabaseType and DatabaseTypeExtended tags of the Data Warehouse Center.

This table shows the entity to Table.tag mapping.

Table 41. The entity to Table.tag mapping

XML tag	Data Warehouse Center tag
EntityProps.Definition	TableDescription
EntityProps.Comment	TableNotes
EntityProps.DB_Owner and Physical_Name	TableFullName
EntityProps.DB_Owner	TableOwner
EntityProps.Physical_Name	TablePhysicalName
ModelProps.Name	DatabaseName
ModelProps.Name	DatabasePhysicalName

This table shows the attribute to Column.tag mapping.

Table 42. The attribute to Column.tag mapping

XML tag	Data Warehouse Center tag
AttributeProps.Name	ColumnName
AttributeProps.Definition	ColumnDescription
AttributeProps.Comment	ColumnNotes
Not applicable	ColumnUserActions
DomainProps.Datatype (strings only)	ColumnLength ^{1, 2}
Key_Group_MemberProps. Key_Group_Position	ColumnKeyPosition
AttributeProps.Attribute_Required, AttributeProps.Null-Option	ColumnAllowsNulls
Domain.Name, DomainProps.ParentDomainName	ColumnDataIsText ^{1, 3}

Metadata mappings

Table 42. The attribute to Column.tag mapping (continued)

XML tag	Data Warehouse Center tag
(native data type)	ColumnNativeDataType ¹
EntityProps.Ordinal - 1	ColumnOrdinalNumber
Not applicable	ColumnOffsetFromZero
ModelProps.Name	DatabaseName
AttributeProps.Order	ColumnPositionNumber

1. Determined by best match.
2. If a value cannot be found, the value is zero.
3. The value is set to Y for CHAR, VARCHAR, LONGVARCHAR, GRAPHIC, VARGRAPHIC, and LONGVARGRAPHIC native data types. Otherwise, the value is set to N.

Related concepts:

- “IBM ERwin 3.5.2 metadata extract program” on page 180

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Mapping ERwin Version 3.5.2 object attributes to Data Warehouse Center tags” on page 236
- “Metadata mappings between the Data Warehouse Center and CWM XML objects and properties” on page 238
- “Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center” on page 233
- “ERwin to Information Catalog Center mapping” in the *Information Catalog Center Administration Guide*
- “Metadata mappings between ERwin Version 4.0 object attributes and Information Catalog Center properties” in the *Information Catalog Center Administration Guide*
- “db2erwinimport command for ERwin 4.0 or later” on page 181

Mapping ERwin Version 3.5.2 object attributes to Data Warehouse Center tags

The following tables show how the main ERwin Version 3.5.2 object attributes correspond to the Data Warehouse Center tags:

This table shows the Database.

Table 43. WarehouseDatabase.tag or SourceDatabase.tag mapping

ERwin	Command line tag	Data Warehouse Center
Diagram Name	NAME	Name of warehouse source or warehouse target
Diagram Author	RESPNSBL	Contact
Database Name	DBNAME	Database name
Database Version	DBTYPE	Database type
Diagram Description	SHRTDESC	Description

This table shows the Table.

Table 44. Table.tag mapping

ERwin	Command line tag	Data Warehouse Center
Table Name	NAME	Table name
Table Name	TABLES	Table name
Database Name	DBNAME	Not applicable
Table Owner	OWNER	Table schema
Table Comment	SHRTDESC	Description

This table shows the Column.

Table 45. Column.tag mapping

ERwin	Command line tag	Data Warehouse Center
Column Name	NAME	Column name
Datatype	NATIVEDT	Datatype
Length	LENGTH	Length
Scale	SCALE	Scale
Null Option	NULLABLE	Allows Nulls (check box)
Position	POSNO	Not applicable
Primary Key	KEYPOSNO	Not applicable
Database Name	DBNAME	Not applicable
Table Owner	OWNER	Not applicable
Table Name	TABLES	Not applicable
Column Comment	SHRTDESC	Description

Related concepts:

- “IBM ERwin 3.5.2 metadata extract program” on page 180

Related tasks:

- “Creating tag language files for the IBM ERwin Metadata Extract program” on page 184

Trillium DDL to Data Warehouse Center metadata mapping

To create the metadata for the source and target files, the Trillium Software System reads the Trillium DDL files. The DDL file is converted to the data types shown in the following table.

Table 46. DDL data types and the equivalent warehouse data types

DDL data types	Warehouse data type
ASCII CHARACTER	CHARACTER(<i>n</i>)
ASCII NUMERIC	
EBCDIC CHARACTER	
EBCDIC NUMERIC	

Metadata mappings

Table 46. DDL data types and the equivalent warehouse data types (continued)

DDL data types	Warehouse data type
Other types	NUMERIC

Note: EBCDIC CHARACTER and EBCDIC NUMERIC data types are supported only if the Trillium Software System is running on the z/OS operating system.

The variable n is the number of characters in the string.

Related concepts:

- “Trillium Batch System JCL files” on page 157
- “Trillium Software System components” on page 155
- “Trillium metadata” on page 156
- “Trillium Batch System user-defined program” on page 158
- “Error handling for Trillium Batch System programs” on page 160

Related tasks:

- “Cleaning name and address data with the Trillium Software System” on page 153
- “Importing Trillium metadata” on page 156

Related reference:

- “Parameters for the Trillium Batch System script or JCL” on page 159

Metadata mappings between the Data Warehouse Center and CWM XML objects and properties

The following table shows how Data Warehouse Center objects map to CWM XML objects, based on the Common Warehouse Metamodel (CWM) Specification V1.0.

Table 47. Equivalent CWM XML objects

Data Warehouse Center objects	Equivalent CWM XML objects
Warehouse source or warehouse target	Catalog + SoftwareSystem("DBM") + Component + Datamanager+Machine
Table	Schema + Table + View
Column or field	Column or Field + SQLSimpleDataType
ForeignKey	ForeignKey
PrimaryKey	PrimaryKey
Column mapping	ClassifierMap + FeatureMap
(Warehouse source or warehouse target) + file	RecordFile + RecordDef + SoftwareSystem("File System")+Component+Datamanager +Machine
Agent Site	Machine + DeployedComponent("Agent Site") + Component + SoftwareSystem("DWC")
Subject	ProcessPackage
Process	WarehouseActivity

Table 47. Equivalent CWM XML objects (continued)

Data Warehouse Center objects	Equivalent CWM XML objects
Step	WarehouseStep + TransformationStep + TransformationTask + Transformation(if without attribute mapping)/TransformationMap(if with attribute mapping)
User-Defined Programs	SoftwareSystem
Program	Component
Parameters	Parameter + DataType
Step Program	Operation + Parameter

The following table shows how Data Warehouse Center properties map to CWM XML properties.

Table 48. Equivalent CWM XML properties

Data Warehouse Center properties	Equivalent CWM XML properties
Description or Notes	Description
Administrator	ResponsibleParty

Related concepts:

- “IBM ERwin 3.5.2 metadata extract program” on page 180

Related tasks:

- “Importing metadata” on page 173
- “Metadata export capabilities” on page 171
- “iwh2imp2 command syntax for ERwin 3.5.2 to import metadata into the Data Warehouse Center” on page 185
- “Creating tag language files for the IBM ERwin Metadata Extract program” on page 184
- “Importing Trillium metadata” on page 156
- “Exporting the metadata to the Data Warehouse Center” on page 207

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 237
- “Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center” on page 233

Metadata mappings

Appendix B. Defining values for warehouse deprecated programs and transformers

Deprecated programs are located in the Warehouse Deprecated Programs folder in the Process Modeler window. Do not use the programs in this folder to create new steps. Instead, use the warehouse programs and transformers for all step types. This appendix provides information about the following deprecated programs:

- Visual Warehouse 5.2 programs
- Data Warehouse Center Version 7.2 Clean Data transformer

Defining values for Visual Warehouse 5.2 compatible programs

This section explains how to use Visual Warehouse 5.2 programs in the Data Warehouse Center.

Defining values for a Visual Warehouse 5.2 DB2 UDB Data Export (VWPEXPT1) program

Use this step to run the Visual Warehouse DB2 UDB Data Export (VWPEXPT1) warehouse program in the Data Warehouse Center. You use the Data Export program to export data from the local DB2 database. It is recommended that you do not use this program to create new steps. Instead, use the data warehouse program.

If the export process generates warnings, the program returns as successfully completed.

This program is available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Solaris Operating Environment

Prerequisites:

Connect a warehouse source to this step in the Process Model window before you define the values for this step subtype. Parameter values for this step subtype will be defined automatically based on your source definition.

Restrictions:

The Column Mapping page is not available for this step.

Procedure:

To define values for a step that uses this program:

1. Open the step notebook.
2. Specify general information about the program.

Defining values for warehouse deprecated programs and transformers

3. On the Parameters page, double-click the **Parameter value** field for the SELECT statement and specify a SELECT statement for the program.
4. Optional: Double-click the **Parameter value** field for the File type modifier string parameter, and type the file type modifier string. For example, the File type modifier string parameter "coldel," indicates that columns must be separated by commas. Do not insert a space between coldel and the comma. Make sure that you enclose this parameter in double quotation marks. Otherwise, the command line processor interprets some characters as file redirection characters.
5. On the Processing Options page, provide information about how your step processes.
6. Click **OK** to save your changes and close the step notebook.

Related tasks:

- "Defining values for a DB2 UDB export utility" on page 122
- "Defining values for the Red Brick SQL Export utility" on page 123

Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Insert (VWPLOADI) program

Use this step to run the Visual Warehouse 5.2 DB2 UDB Data Load Insert (VWPLOADI) warehouse program in the Data Warehouse Center. You use VWPLOADI to load data from a flat file into a DB2 table, appending to existing data. It is recommended that you do not use this program to create new steps. Instead, use the warehouse program.

The Visual Warehouse 5.2 DB2 UDB Load Insert warehouse program extracts the following step and warehouse source parameter values from the Process Model window and your step definition:

- The flat file that is selected as a source for the step. The step must have only one selected source file. The source file must contain the same number and order of fields as the target tables. Only delimited ASCII (ASCII DEL) source files are supported. For information about the format of delimited files, see the *DB2 Command Reference*.
- The warehouse target database name. You must have either SYSADM or DBADM authorization to the DB2 database. The DB2 UDB Load Insert program does not support multinode databases. For multinode databases, use Load flat file into DB2 ESE (VWPLDPR) for DB2 UDB Enterprise Server Edition.
- The user ID and password for the warehouse target.
- The target table that is defined for the step.

These parameters are predefined. You do not specify values for these parameters. Additionally, the step passes other parameters for which you provide values. Before the program loads new data into the table, it exports the table to a backup file, which you can use for recovery.

Recommendation: Create the target table in its own private DB2 table space. Any private table space that you create will be used by default for all new tables that do not specify a table space. If processing fails, DB2 might put the whole table space in hold status, making the table space inaccessible. To avoid this hold problem, create a second private table space for steps that do not use the load programs.

Defining values for warehouse deprecated programs and transformers

If the warehouse program detects a failure during processing, the table will be emptied. If the load generates warnings, the program returns as successfully completed.

The warehouse program does not collect database statistics. Run the DB2 UDB RUNSTATS program after a sizable load is complete.

This warehouse program is available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Solaris Operating Environment

Prerequisites:

Connect the step to warehouse source and a warehouse target in the Process Model window.

Restrictions:

- The Data Warehouse Center definition for the warehouse agent site that is running the program must include a user ID and password. The DB2 load utility cannot be run by a user named SYSTEM. Be sure to select the same warehouse agent site in the warehouse source and the warehouse target for the step using the program. The database server does not need to be on the agent site. However, the source file must be on the database server. Specify the fully qualified name of the source files as defined on the DB2 server.
- The Column Mapping page is not available for this step.

Procedure:

To create a tablespace:

```
CREATE TABLESPACE tablespace-name MANAGED BY SYSTEM USING ('d:/directory')
```

where *directory* is the directory that is to contain the databases. DB2 creates the directory for you.

To define a values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program.
3. Double-click the **Parameter value** field for the Backup file name parameter, and type the fully qualified name of the backup file.
4. Double-click the **Parameter value** field for the File type modifier string parameter, and type the file type modifier string file type modifier string. For example, the File type modifier string parameter "coldel," indicates that columns must be separated by commas. Do not insert a space between coldel and the comma. Make sure that you enclose this parameter in double quotation marks. Otherwise, the command line processor interprets some characters as file redirection characters.
5. On the Processing Options page, provide information about how your step processes.
6. Click **OK** to save your changes and close the step notebook.

Defining values for warehouse deprecated programs and transformers

Related tasks:

- “Defining values for a DB2 Universal Database load utility” on page 124
- “Defining a DB2 for z/OS load utility” on page 130

Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Replace (VWPLOADR) program

Use this step to run the Visual Warehouse 5.2 DB2 Data Load Replace (VWPLOADR) warehouse program in the Data Warehouse Center. You use VWPLOADR to load data from a flat file into a DB2 table, replacing existing data. It is recommended that you do not use this program to create new steps. Instead, use the warehouse program.

The Visual Warehouse 5.2 DB2 UDB Load Replace warehouse program extracts the following step and warehouse source parameter values from the Process Model window and your step definition:

- The flat file that is selected as a source for the step. The step must have only one source file selected. The source file must contain the same number and order of fields as the target tables. Only delimited ASCII (ASCII DEL) source files are supported.
- The warehouse target database name. You must have either SYSADM or DBADM authorization to the DB2 database. This program does not support multinode databases. For multinode databases, use the Load flat file into DB2 ESE (VWPLDPR) program for DB2 UDB Enterprise Server Edition.
- The user ID and password for the warehouse target.
- The target table that is defined for the step.

These parameters are predefined. You do not specify values for these parameters.

Recommendation: Create the target table in its own private DB2 table space. Any private table space that you create will be used for all new tables that do not specify a table space. If processing fails, DB2 might put the whole table space in hold status, making the table space inaccessible. To avoid this hold problem, create a second private table space for steps that do not use the load programs.

If the warehouse program detects a failure during processing, the table is emptied. If the load generates warnings, the program returns as successfully completed.

The DB2 UDB Load Replace program collects database statistics during the load, so you do not need to run the DB2 UDB RUNSTATS (VWPSTATS) program after this program.

This warehouse program is available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Solaris Operating Environment

Prerequisites:

Connect the step to warehouse source and a warehouse target in the Process Model window.

Defining values for warehouse deprecated programs and transformers

Restrictions:

- The Data Warehouse Center definition for the agent site that is running the program must include a user ID and password. The DB2 load utility cannot be run by a user named SYSTEM. Be sure to select the same agent site in the warehouse source and the warehouse target for the step that uses the warehouse program. The database server does not need to be on the agent site. However, the source file must be on the database server. Specify the fully qualified name of the source files as defined on the DB2 server.

Procedure:

To create a table space:

```
CREATE TABLESPACE tablespace-name MANAGED BY SYSTEM USING ('d:/directory')
```

where *directory* is the directory that is to contain the databases. DB2 creates the directory.

To define a values for a step that uses this program:

1. Open the step notebook.
2. Specify general information about the warehouse program.
3. Optional: On the Parameters page, double-click the **Parameter value** field for the File type modifier string parameter, and type the file type modifier string. For example, the File type modifier string parameter "coldel," indicates that columns must be separated by commas. Do not insert a space between coldel and the comma. Make sure that you enclose this parameter in double quotation marks. Otherwise, the command line interprets some characters as file redirection characters.
4. On the Processing Options page, provide information about how your step processes.
5. Click **OK** to save your changes and close the step notebook.

File type modifiers

The file type modifier can use the following modifiers:

Table 49. File type modifiers

Modifier	Description
Chardel <i>x</i>	<i>x</i> is a single-character string delimiter. The default value is a double quotation mark ("). The character that you specify is used in place of double quotation marks to enclose a character string. You can specify a single quotation mark (') as a character string delimiter as follows: Modified by chardel "
Coldel <i>x</i>	<i>x</i> is a single-character column delimiter. The default value is a comma (.). The character that you specify is used instead of a comma to signal the end of a column. Do not insert a space between coldel and the comma. Enclose this parameter in double quotation marks. Otherwise, the command line processor interprets some characters as file redirection characters. In the following example, coldel ; causes the export utility to interpret any semicolon (;) it encounters as a column delimiter: Db2 'export to temp of del modified by coldel; select * from staff where dept = 20"
Dateisiso	Dateisiso causes all date data values to be exported in ISO format.

Defining values for warehouse deprecated programs and transformers

Table 49. File type modifiers (continued)

Modifier	Description
Decplusblank	Decplusblank causes positive decimal values to be prefixed with a blank space instead of a plus sign (=). The default action is to prefix positive decimal values with a plus sign.
Decpt <i>x</i>	<i>x</i> is a single-character substitute for the period as a decimal point character. The default value is a period (.). The character that you specify is used in place of a period as a decimal point character.

Defining values for a Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program

Use this step to run the Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program in the Data Warehouse Center. This step runs the DB2 UDB REORG utility on a target table. It is recommended that you do not use this program to create new steps. Instead, use the data warehouse program.

This program is available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Solaris Operating Environment

You schedule this step to run on the target table of a process, after the process completes.

The Visual Warehouse 5.2 DB2 UDB REORG program extracts the following step and warehouse source parameter values from the Process Model window and your step definition:

- The warehouse target database name
- The user ID and password for the warehouse target
- The target table that is defined for the step

These parameters are predefined. You do not specify values for these parameters.

Prerequisites:

In the Process Model window, draw a data link from the step to the warehouse target.

Procedure:

To define a values for a step that uses this program:

1. Open the step notebook.
2. Specify general information about the warehouse program.
3. On the Processing Options page, provide information about how your step processes.
4. Click **OK** to save your changes and close the step notebook.

Defining values for a Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) program

Use this step to run the Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) warehouse program in the Data Warehouse Center. It is recommended that you do not use this program to create new steps. Instead, use the data warehouse program.

This program is available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Solaris Operating Environment

This step runs the DB2 UDB RUNSTATS utility on a target table. You schedule this step to run on the target table of a process, after the process completes.

The Visual Warehouse 5.2 DB2 UDB RUNSTATS warehouse program extracts the following step and warehouse source parameter values from the Process Model window and your step definition:

- The warehouse target database name
- The user ID and password for the warehouse target
- The target table that is defined for the step

These parameters are predefined. You do not specify values for these parameters.

Prerequisites:

In the Process Model window, draw a data link from the step to the warehouse target.

Procedure:

To define a values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program.
3. On the Processing Options page, provide information about how your step processes.
4. Click **OK** to save your changes and close the step notebook.

Defining values for a Visual Warehouse 5.2 Load flat file into DB2 Enterprise Server Edition (VWPLDPR) program (AIX)

Use this step to run the Visual Warehouse 5.2 Load flat file into DB2 Enterprise Server Edition (AIX only) (VWPLDPR) program in the Data Warehouse Center. This step loads data from a delimited flat file into a DB2 Universal Database Enterprise Server Edition database, replacing existing data. It is recommended that you do not use this program to create new steps. Instead, use the warehouse program.

This program is available for AIX.

Defining values for warehouse deprecated programs and transformers

The VWPLDPR program performs the following steps when it is loading data to a parallel database:

1. Connects to the target database.
2. Acquires the target partitioning map for the database.
3. Splits the input file so each file can be loaded on a database partition.
4. Runs remote load on all database partitions.

If the load step fails on any database partition, the VWPLDPR program does the following:

1. Builds an empty load data file for each database partition.
2. Loads the empty data files.

The VWPLDPR program extracts the following step and warehouse source parameter values from the Process Model window and your step definition:

- The flat file that is selected as a source for the step. The step must have only one source file selected. Only delimited (DEL) files are supported. The input file and the split files must be on a file system that is shared by all database partitions involved in the database load. The shared file system must be mounted on the same directory on all database partitions. The directory must be large enough to contain the input file before and after the file is split.
- The warehouse target database name.
- The user ID and password for the warehouse target.
- The target table that is defined for the step.

These parameters are predefined. You do not specify values for these parameters. Additionally, there are a number of parameters that you must provide values for.

The Load flat file into DB2 Extended Enterprise Edition program does not run the DB2 RUNSTATS utility after the load. If you want to automatically run the RUNSTATS utility after the load, add a step to your process that runs RUNSTATS .

Recommendation: Create the target table in its own private DB2 table space. Any private table space that you create will be used for all new tables that do not specify a table space. If processing fails, DB2 might put the whole table space in hold status, making the table space inaccessible. To avoid this hold problem, create a second private table space for steps that do not use the load programs.

Prerequisites:

Before you use this warehouse program, you must be familiar with parallel system concepts and parallel load.

Restrictions:

This program runs only on AIX.

Procedure:

To create a table space:

```
CREATE TABLESPACE tablespace-name MANAGED BY SYSTEM USING (d:/directory')
```

where *directory* is the directory that is to contain the databases. DB2 creates the directory.

Defining values for warehouse deprecated programs and transformers

To define a values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the program.
3. On the Parameters page, provide values for the following parameters:
 - a. Double-click the **Parameter value** field for the Column delimiter parameter, and type the column delimiter. The most common values are a comma (,) or a semicolon (;).
 - b. Double-click the **Parameter value** field for the string parameter, and type the string parameter. The most common value is a double quotation mark ("), entered as "\".
 - c. Double-click the **Parameter value** field for the Decimal delimiter parameter, and type the decimal delimiter. The most common values are a period (.) or a comma (,).
 - d. Double-click the **Parameter value** field for the local, non-shared file directory parameter, and type the local, non-shared file directory.
 - e. Double-click the **Parameter value** field for the Path name and prefix for the parameter, and type the path name and prefix for the split files. The name of each file will consist of the prefix plus a numeric identifier.
 - f. Double-click the **Parameter value** field for the Partition key parameter, and type a parameter for each partition key. The partition key must be in the format used by the db2split database utility. The format generally is as follows: *col1,1,,,N,integer* followed by *col3,3,,,5N,character*
4. On the Processing Options page, provide information about how your step processes.
5. Click **OK** to save your changes and close the step notebook.

Defining values for the Data Warehouse Center Version 7.2 Clean Data transformer

This section explains how to define values for the Data Warehouse Center Version 7.2 Clean Data transformer.

Data Warehouse Center Version 7.2 Clean Data transformer

Use the Data Warehouse Center Version 7.2 Clean Data transformer to perform rules-based find and replace operations on a table. The transformer finds values that you specify in the data columns of the source table that your step accesses. Then the transformer updates corresponding columns with replacement values that you specify in the table to which your step writes. You can select multiple columns from the input table to carry over to the output table. The Data Warehouse Center Version 7.2 Clean Data transformer does not define rules or parameters for the carry over columns. It is recommended that you do not use this transformer to create new steps. Instead, use the Clean Data transformer located under warehouse transformers in the Process Model window.

Use the Data Warehouse Center Version 7.2 Clean Data transformer to clean and standardize data values after load or import, as part of a process. Do not use this transformer as a general-purpose data column editor.

You can use the Data Warehouse Center Version 7.2 Clean Data transformer to perform the following tasks:

Defining values for warehouse deprecated programs and transformers

- Replace values in selected data columns that are missing, not valid, or inconsistent with appropriate substitute values
- Remove unsuitable data rows
- Clip numeric values
- Perform numeric discretization
- Remove excess white space from text
- Copy columns from the source table to the target table

You can use the Data Warehouse Center Version 7.2 Clean Data transformer only if your source table and target table are in the same database. The source table must be a single warehouse table. The target table is the default target table.

You can choose to ignore case and white space when locating strings, and you can specify a tolerance value for numeric data.

You can make changes to the step only when the step is in development mode.

Each clean transformation that you specify uses one of four clean types:

Find and replace

Performs basic find and replace functions.

Discretize

Performs find and replace functions within a range of values.

Clip Performs find and replace functions within a range of values or outside of a range of values.

Carry over

Specifies columns in the input table to copy to the output table.

Prerequisite: Before you can use the Data Warehouse Center Version 7.2 Clean Data transformer, you must create a rules table for your clean type. A rules table designates the values that the Data Warehouse Center Version 7.2 Clean Data transformer will use during the find and replace process. The rules table must be in the same database as the source table and target table.

Related concepts:

- “Rules tables for Data Warehouse Center Version 7.2 Clean Data transformer” on page 250
- “Clean Data transformer” on page 143

Related tasks:

- “Defining a Data Warehouse Center Version 7.2 Clean Data transformer” on page 252

Rules tables for Data Warehouse Center Version 7.2 Clean Data transformer

This topic describes rules tables for the Data Warehouse Center Version 7.2 Clean Data transformer. It is recommended that you do not use this transformer to create new steps. Instead, use the Clean Data transformer located under warehouse transformers in the Process Model window. At a minimum, a rules table must contain at least two columns. One column contains find values. The other column contains replace values. The rows in each column correspond to each other.

Defining values for warehouse deprecated programs and transformers

For example, Column 1 and Column 2 in a rules table have the values shown here:

Table 50. Rules table example

Column 1	Column 2
Desk	Chair
Table	Lamp

Suppose that Column 1 contains the find values, and Column 2 contains the replace values. When you run the step, the Data Warehouse Center Version 7.2 Clean Data transformer searches your source column for the value Desk. Wherever it finds the value Desk, it writes the value Chair in the corresponding field of the target column.

The Data Warehouse Center Version 7.2 Clean Data transformer copies values that are not listed in the find column directly to the target table. In the example, the value Stool is not listed in the column that contains the find values. If the selected source column contains the value Stool, the Clean transformer will write Stool to the corresponding field in the target column.

The following table describes the columns that must be included in the rules table for each clean type:

Table 51. Required columns

Clean type	Minimum number of columns in rules table	Purpose of columns
Find and replace	2 - 3	The first column contains find values. The second column contains replace values. The third column, if specified, contains rule-ordering information and is not required.
Discretize	3 - 4	The first column contains the low end of the find value range. The second column contains the high end of the find value range. The third column contains the replacement value. The fourth column, if specified, contains rule-ordering information and is not required.

Defining values for warehouse deprecated programs and transformers

Table 51. Required columns (continued)

Clean type	Minimum number of columns in rules table	Purpose of columns
Clip	3 - 5	<p>The first column contains the low end of the find value range.</p> <p>The second column contains the high end of the find value range.</p> <p>The third column contains the low end of the replacement value.</p> <p>The fourth column contains the high end of the replacement value.</p> <p>The fifth column, if specified, contains rule-ordering information and is not required.</p> <p>Tip: When you use the Clip clean type, you can run a find and replace procedure that searches outside of value ranges.</p>
Carry over	None allowed	Rule table not used.

You can reorder the output columns using the Properties notebook for the step. You can change column names on the Column Mapping page of the Properties notebook for the step.

Defining a Data Warehouse Center Version 7.2 Clean Data transformer

You can use the Data Warehouse Center Version 7.2 Clean Data transformer in the Data Warehouse Center. It is recommended that you do not use this transformer to create new steps. Instead, use the Clean Data transformer located under warehouse transformers in the Process Model window.

Procedure:

To define a Data Warehouse Center Version 7.2 Clean Data transformer:

1. Open the step notebook
2. Specify general information for your step
3. On the Parameters page, define the clean transformations that you want performed when the step runs.
4. On the Column Mapping page, map the source columns to target columns.
5. On the Processing Options page, provide information about how your step processes.

Related concepts:

- “Data Warehouse Center Version 7.2 Clean Data transformer” on page 249
- “Rules tables for Data Warehouse Center Version 7.2 Clean Data transformer” on page 250

Related reference:

- “Warehouse deprecated programs” on page 94

Appendix C. Data Warehouse Center environment structure

The information in this appendix explains how the Data Warehouse Center is structured and how it interacts with the operating system.

Data Warehouse Center environment variables

In the following table, the drive letter C indicates a local drive.

Table 52. Data Warehouse Center environment variables

Environment variable	Is added or modified to include:
PATH (used to access the Data Warehouse Center code)	C:\Program Files\SQLLIB\BIN
LOCPATH (used by the Data Warehouse Center Host Adapter Client)	C:\Program Files\SQLLIB\ODBC32\LOCALE
VWS_TEMPLATES	C:\Program Files\SQLLIB\TEMPLATES
VWS_LOGGING	C:\Program Files\SQLLIB\LOGGING
VWSPATH	C:\Program Files\SQLLIB
INCLUDE	C:\Program Files\SQLLIB\TEMPLATES\INCLUDE
VWS_LOCALE	Is not added to or modified during install

To fix Unicode character conversion errors, you can use the VWS_LOCALE environment variable to override the ICU converter used by the DB2 Warehouse Manager. You can use the ICU Converter Explorer tool, or an equivalent tool, to identify the ICU converter names that can be used by the DB2 Warehouse Manager and specified by the VWS_LOCALE environment variable to override the default behavior. You can find the ICU Converter Explorer tool at the following Web site: <http://oss.software.ibm.com/cgi-bin/icu/convexp>.

Related concepts:

- “Data Warehouse Center load and export programs” on page 121

Related reference:

- “Data Warehouse Center registry updates” on page 253

Data Warehouse Center registry updates

The following values are added to the Windows registry in the following key:
HKEY_LOCAL_MACHINE\Software\IBM\DB2\DataWarehouseCenter\ServiceParms

Value name	Value data
Database name	<i>Control DB name</i>
Log Directory	<i>disk:\dir\</i>
Password	<i>password</i>

Data Warehouse Center environment structure

Qualifier	<i>table qualifier</i>
Userid	<i>DB2 user ID</i>

Related concepts:

- “Data Warehouse Center configuration” on page 221

Related reference:

- “Data Warehouse Center configuration files” on page 254

Data Warehouse Center configuration files

The following updates are made to TCP/IP configuration information:

TCP/IP SERVICES:

```
vwkernel    11000/tcp
vwlogger    11002/tcp
vwd         11001/tcp
```

Related concepts:

- “Data Warehouse Center configuration” on page 221

Related reference:

- “Data Warehouse Center environment variables” on page 253
- “Data Warehouse Center registry updates” on page 253

Notices

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

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

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

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

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

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

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

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

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

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

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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

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

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

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

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

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

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

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

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

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM	iSeries
AISPO	LAN Distance
AIX	MVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	NetView
BookManager	OS/390
C Set++	OS/400
C/370	PowerPC
CICS	pSeries
Database 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RISC System/6000
DataRefresher	RS/6000
DB2	S/370
DB2 Connect	SP
DB2 Extenders	SQL/400
DB2 OLAP Server	SQL/DS
DB2 Information Integrator	System/370
DB2 Query Patroller	System/390
DB2 Universal Database	SystemView
Distributed Relational Database Architecture	Tivoli
DRDA	VisualAge
eServer	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
IBM	WebSphere
IMS	WIN-OS/2
IMS/ESA	z/OS
	zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

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

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

Index

A

- administration privilege 20
- administrative interfaces
 - adding or deleting from a warehouse 218
- agent site configurations 17
- agent sites
 - defining 18
 - defining for user-defined programs 190
- agents
 - accessing DB2 Relational Connect sources
 - zSeries 37
 - accessing local and remote databases
 - iSeries 35
 - adding or deleting from a warehouse 218
 - changing 44
 - data warehouse 2
 - described 2, 11
 - requirements for accessing remote databases
 - iSeries 34
 - running as user processes
 - Windows NT, Windows 2000, Windows XP 230
 - starting
 - iSeries 12
 - Windows NT, Windows 2000, or Windows XP 12
 - starting in the foreground
 - z/OS 13
 - stopping 16
 - AIX, Solaris Operating Environment, Linux 16
 - Windows NT, Windows 2000, Windows XP 15
 - zSeries 16
 - support for DB2 sources 26
 - support for file sources 63
 - support for non-DB2 sources 40
 - verifying running status
 - iSeries 12
 - zSeries 14
 - verifying that the agent has started
 - iSeries 12
 - verifying the version
 - AIX, Solaris Operating Environment, Linux 44
- analysis of variance
 - defining a warehouse transformer 163
- ANOVA transformer
 - defining 163
- Apply program
 - tracing errors 232

B

- build-time errors 228
- BVBESTATUS tables
 - creating 78
 - using with DB2 Relational Connect 79

C

- CA/400
 - defining a file extension 129
- Calc with calc rules (ESSCALC2) program
 - description 92
 - running calculations on OLAP Integration Server data 224
- Calculate Statistics transformer 164
- Calculate Subtotals transformer 164
- calculated columns
 - defining 119
- calculation scripts 223
- change aggregate steps, defining 139
- changing your configuration 217
- chi-square transformer 166
- Clean Data transformer
 - clean types for DWC 7.2
 - described 249
 - defining for DWC 7.2 252
 - description 143
 - rules tables described 148
- Client Access/400
 - defining a file extension 129
- code transformation 114
- column mapping
 - description, for Data Warehouse Center 98
- command scripts
 - loading for the DB2 OLAP Integration Server 206
- common warehouse metadata (cwm)
 - exporting 171
- common warehouse metamodel (cwm) metadata
 - importing 173
- common warehouse metamodel XML (CWM XML) metadata
 - mapping metadata 238
- composite summary steps, creating 118
- configuration files
 - Data Warehouse Center 254
- configuring Data Warehouse Center
 - changing 217
- connectivity
 - accessing a DB2 DRDA source
 - Windows NT, Windows 2000, Windows XP 26
 - accessing a DB2 for z/OS warehouse database 76
 - accessing a DB2 source
 - Windows NT, Windows 2000, Windows XP 26

connectivity (continued)

- accessing a Sybase Adaptive Server source, AIX 45
- accessing a z/OS or VM file source
 - Solaris Operating Environment, Linux 69
- accessing an Oracle 8 source
 - AIX 45
 - Windows NT, Windows 2000, Windows XP 41
- accessing DB2 UDB sources 31
- accessing Informix 9.2 sources (with client)
 - Windows NT, Windows 2000, Windows XP 42
- accessing Informix 9.2 sources (without client)
 - AIX 46
 - Windows NT, Windows 2000, Windows XP 42
- accessing Microsoft SQL Server sources
 - AIX 47
 - Windows NT, Windows 2000, Windows XP 42
- accessing remote file sources
 - Solaris Operating Environment, Linux 70
 - Windows NT, Windows 2000, Windows XP 67
- accessing Sybase Adaptive Server sources
 - Windows NT, Windows 2000, Windows XP 41
- accessing z/OS or VM file sources
 - Windows NT, Windows 2000, Windows XP 65
- connecting to DB2 DRDA sources 33
- establishing zSeries agents 36
 - requirements
 - accessing remote databases with the iSeries warehouse agent 34
 - connecting the warehouse agent to the warehouse server 11
- setting up for a DB2 Universal Database source
 - OS/2 30
 - Solaris Operating Environment, Linux 28
- setting up for a DRDA source
 - OS/2 30
 - Solaris Operating Environment, Linux 29
- setting up for a Microsoft SQL Server source
 - Solaris Operating Environment, Linux 47
- setting up for DB2 UDB sources
 - AIX 27
- setting up for DRDA sources
 - AIX 28

- constants
 - adding to an expression 120
- control database
 - backing up 217
 - initializing during installation 218, 219
- control tables
 - creating
 - for replication 138
- Copy File using FTP warehouse program
 - accessing data with 65
 - described 91
 - moving files 133
- correlation coefficient 166
- Correlation transformer 166
- covariance 166
- creating
 - target tables with DB2 Relational Connect 80
- CWM (common warehouse metamodel)
 - XML metadata
 - mapping to Data Warehouse Center objects 238

D

- data
 - copying between DB2 utilities 131
 - defining transformation and movement 88
 - filtering 117
 - inserting 111
 - loading into a multidimensional cube 205
 - selecting 111
- data export
 - with ODBC to file utility 122
 - with ODBC to file warehouse program 91
- data modeling
 - IBM ERwin Metadata Extract Program 180
- data movement and transformation
 - using replication steps 139
- Data Transaction Services support
 - Data Warehouse Center 197
- data transformation and movement
 - incremental commit described 112
 - publishing warehouse metadata to an information catalog 174
 - replication described 135
 - replication programs defined 93
- Data Warehouse Center
 - administrative interface
 - starting 7
 - starting (AIX) 8
 - configuration files 254
 - configuring properties 221
 - creating a status table 78
 - defining a DB2 Enterprise Server Edition database 78
 - environment variables 253
 - logger, starting 9
 - metadata
 - mappings with the DB2 OLAP Integration Server 233
 - OLE DB Support 197

- Data Warehouse Center (*continued*)
 - registry changes
 - Windows NT, Windows 2000, Windows XP 253
 - security 20
 - server, starting 9
 - stopping services
 - Windows NT, Windows 2000, Windows XP 217
- data warehouse objects 2
- data warehousing
 - defined 1
 - informational data 1
 - operational data 1
- database directory entries
 - manipulating remote entries 36
- date fields
 - changing date formats 152
 - specifying the input format 153
 - specifying the output format 153
- date formats
 - changing 152
- DB2 application heapsize
 - exporting metadata 171
 - importing metadata 173
 - publishing Data Warehouse Center metadata 175
- DB2 Connect
 - accessing DB2 for iSeries databases 75
- DB2 Connect gateway sites
 - setting up
 - iSeries 75
 - Windows NT, Windows 2000, Windows XP 33
- DB2 database definitions
 - changing to an ERwin source 186
- DB2 DRDA sources
 - accessing 31
 - defining privileges 32
 - setting up connectivity AIX 28
- DB2 Enterprise Server Edition
 - databases
 - defining privileges 77
 - defining to the Data Warehouse Center 78
- DB2 for iSeries
 - Data Load Insert utility
 - defining 125
 - Data Load Insert warehouse program
 - using trace files 128
 - Data Load Replace utility
 - accessing trace files with CA/400 129
 - defining 126
 - Data Load Replace warehouse program
 - using trace files 128
- databases
 - accessing with DB2 Connect 75
- Load Insert utility
 - accessing a trace file with CA/400 129
- Load utilities
 - viewing trace files 129
- modstring parameters described 127

- DB2 for iSeries (*continued*)
 - sources
 - defining privileges 32
 - warehouses
 - defining privileges 74
- DB2 for VM
 - setting up the DB2 Connect gateway site 33
- DB2 for VM sources
 - accessing 31
 - defining privileges 32
- DB2 for VSE
 - setting up the DB2 Connect gateway site 33
- DB2 for VSE sources
 - accessing 31
 - defining privileges 32
- DB2 for z/OS
 - connecting to a warehouse 76
 - Load warehouse program 130
 - REORG utility 211
 - RUNSTATS utility 215
 - setting up a warehouse 75
 - utilities
 - defining values 211
 - parameters defined 211
 - warehouses
 - defining privileges 76
 - defining to the Data Warehouse Center 76
- DB2 OLAP Integration Server
 - creating a load script 205
 - creating an outline 204
 - exporting metadata to Data Warehouse Center 207
 - loading the command script 206
 - metadata mappings with the Data Warehouse center 233
 - populating a multidimensional cube on a schedule 209
 - programs 92
- DB2 Relational Connect
 - accessing source data 79
 - with the zSeries warehouse agent 37
 - creating a server mapping and a table nickname 60
 - creating target tables 80
 - defining source tables 61
 - defining warehouse sources 60
 - moving a target table 81
 - updating an existing table in a remote database 82
- DB2 sources
 - accessing DB2 UDB sources 31
 - defining 38
 - defining privileges 31, 32
 - warehouse agent support 26
- DB2 Universal Database
 - export utility 122
 - load utility 124
 - REORG utility 211
 - RUNSTATS utility 215
- DB2 Universal Database sources
 - accessing 31
 - setting up connectivity AIX 27

- DB2 Universal Database sources
 - (continued)
 - setting up connectivity for
 - OS/2 30
 - Solaris Operating Environment, Linux 28
 - Windows NT, Windows 2000, Windows XP 26
- DB2 Universal Database warehouses
 - defining privileges 73
- DB2 utilities
 - copying data between utilities 131
 - Defining a DB2 for z/OS REORG utility
 - DB2 UDB or DB2 for z/OS 211
 - defining a DB2 UDB REORG utility 211
 - described 91
- Default calc (ESSCALC1) warehouse program
 - running calculations on OLAP Integration Server data 224
- Default calc program
 - defined 92
- default warehouse user
 - defined 20
- defining
 - SQL 111
- delimited identifier
 - described 37
- deprecated programs
 - described 94
- deprecated warehouse programs
 - defining a Visual Warehouse 5.2 DB2 UDB Data Export program 241
 - defining a Visual Warehouse 5.2 DB2 UDB Data Load Insert (VWPLOADI) program 242
 - defining a Visual Warehouse 5.2 DB2 UDB Data Load Replace (VWPLOADR) 244
 - defining a Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program 246
 - defining a Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) program 247
 - defining a Visual Warehouse 5.2 Load flat file into DB2 Enterprise Extended Edition
 - AIX 247
 - described 88
 - DWC 7.2 Clean Data transformer
 - described 249
 - file type modifiers for Visual Warehouse 5.2 programs 245
- developing steps 100
- development mode 100
- dimension tables 116
- Distributed Relational Database Architecture (DRDA)
 - accessing sources 31
 - setting up connectivity to sources
 - AIX 28
 - OS/2 30
- DRDA sources
 - accessing 31

- DRDA sources (continued)
 - z/OS 33
 - setting up connectivity
 - OS/2 30
 - Solaris Operating Environment, Linux 29
 - Windows NT, Windows 2000, Windows XP 26
- DTS
 - Data Warehouse Center 197
- DTS packages
 - creating views 198
- DWC 7.2 Clean Data transformer
 - defining 252
 - described 249
 - rules tables described 250

E

- environment variables
 - updating
 - Data Warehouse Center 253
 - z/OS 19
- ER1 files 185
- error handling
 - Apply program 232
 - IBM ERwin metadata extract program 186
 - Informix 9.2 (without client) 49
- error messages
 - defined for MQXMLXF user-defined programs 178
 - for steps and processes 109
- errors
 - viewing build-time errors 228
 - viewing log entries 228
 - viewing run-time errors 227
- ERwin metadata
 - importing 185
 - merging with existing database data 185
- ERwin metadata extract program
 - creating a tag language file 184
 - error handling 186
 - starting 180
 - writing to ER1 files 185
- ERwin object attributes
 - mapping to Data Warehouse Center tags 236
- ERwin tag language files
 - exporting 186
- event viewer, warehouse log files 227
- examples
 - cataloging local and remote database names, iSeries 36
 - creating a server mapping and table nickname for DB2 Relational Connect sources 60
 - external trigger program 103
 - job step containing a SYSTEM DD statement 158
 - odbc.ini file entry 70
 - passing parameters to user-defined programs 192
 - specifying pre-defined parameters for user-defined programs 193
 - star join 116

- examples (continued)
 - star schema 116
- export utilities 121
- expressions
 - adding a constant 120
 - adding a function 120
- external trigger program
 - example 103
 - messages 105
 - running steps 102
 - starting 103

F

- fact tables 116
- feedback file 194
- file programs 91
- file sources
 - accessing a z/OS or VM file source from AIX 67
 - accessing local files
 - AIX 68
 - Solaris Operating Environment, Linux 69
 - accessing with Network File System protocol 71
 - cataloging a Microsoft Excel source 53
 - defining to the Data Warehouse Center 63
 - example of an .odbc.ini file entry 70
 - warehouse agent support 63
- file type modifiers
 - Visual Warehouse 5.2 programs 245
- files
 - accessing local files as warehouse sources 66
 - accessing with Copy file using FTP 65
 - accessing with FTP 64
- filtering
 - data, with search conditions 118
- Fisher-F distributions 163
- flat files
 - accessing with Copy file using FTP 65
 - accessing with FTP 64
 - loading into an OLAP Server database 224
- foreign keys
 - description 85
 - star joins 116
- FormatDate transformer 152
- Free text data load program 92
- Free text data load warehouse program 224
- FTP utility
 - accessing files 64
- full outer join 114
- functions
 - adding to an expression 120

G

- Generate Key Table transformer
 - defining 149

Generate Period Table transformer
described 149
goodness-of-fit test 166

I

IBM ERwin metadata extract program
error handling 186
IBM ERwin Metadata Extract program
starting 180
IBM ERwin Metadata Extract
Program 180
incremental commit
described 112
Information Catalog Center
populating with warehouse
metadata 174
Informix 9.2
accessing as a warehouse source (with
client)
Windows NT, Windows 2000,
Windows XP 42
accessing as a warehouse source
(without client)
AIX 46
Windows NT, Windows 2000,
Windows XP 42
configuring an ODBC driver (without
client)
Windows NT, Windows 2000,
Windows XP 48
Informix 9.2 (without client)
error handling 49
inner join 114
inserting data 111
instances
running a 64-bit DB2 instance 86
Invert Data transformer
defining 150
iSeries agent
requirements for accessing remote
databases 34
iSeries agent daemon
verifying running status 12
iSeries databases
accessing with DB2 Connect 75
iSeries warehouse agent
accessing local and remote
databases 35
cataloging local and remote database
name 36
IWH2LOG.LOG 232
IWH2LOGC.LOG 232
IWH2SERV.LOG 232

J

JCL files
using with Trillium Batch system 157
Job steps, example with a SYSTERM DD
statement 158
joins
defining 112
described 114
removing 113

joins (*continued*)
types
star 116

K

key columns
defining Generate Key Table
transformers 149
keys
foreign 116
primary 116

L

left outer join 114
Load data from a file
with load rules (ESSDATA2)
warehouse program
into an OLAP Server
database 224
with load rules program
described 92
without load rules (ESSDATA4)
warehouse program
into an OLAP Server
database 224
without load rules program
described 92
Load data from SQL table
with load rules (ESSDATA3)
warehouse program
into an OLAP Server
database 224
with load rules program
described 92
load replace utility
described
DB2 for iSeries 91
load utility
described 121
DB2 for z/OS 91
loading
data
defining a Data Load Insert utility
for iSeries 125
local databases
cataloging for the iSeries warehouse
agent 36
local file sources
accessing
AIX 68
Solaris Operating Environment,
Linux 69
Windows NT, Windows 2000,
Windows XP 66
logging
error logging for MQSeries 179
error logging for warehouse programs
and transformers 231
running a Data Warehouse Center
component trace 230
tracing warehouse components 228
viewing build-time errors 228
viewing log entries 228
viewing run-time errors 227

M

messages
error return codes for name and
address cleansing defined 161
external trigger program 105
handling Trillium Batch System
errors 160
importing MQSeries messages 176
MQSeries 175
metadata
exporting 171
exporting from DB2 OLAP Integration
Server to the Data Warehouse
Center 207
exporting from the DB2 OLAP
Integration Server 208
exporting to a tag language file 172
extracting 180
importing 173
importing ERwin metadata 185
importing from Trillium Software
System 156
importing XML metadata 176
increasing heapsize for
publishing 175
merging ERwin metadata with
existing metadata 185
publishing warehouse metadata 174
publishing warehouse schema
metadata 209
Trillium metadata described 156
metadata mappings
Data Warehouse Center to CWM
XML 238
DB2 OLAP Integration Server and the
Data Warehouse center 233
ERwin to Data Warehouse
Center 236
Trillium DDL to Data Warehouse
Center 237
Microsoft Access
accessing as a warehouse source 43
cataloging a warehouse target for 52
cataloging as a source 51
defining a warehouse 50
importing table definitions into the
Data Warehouse Center 51
Microsoft Excel
accessing as a warehouse source 43,
53
cataloging a warehouse target for 55
cataloging in ODBC 53
creating a warehouse target for 54
creating named tables for 54
defining warehouse sources and
targets 55
Microsoft SQL Server
accessing as a warehouse source
AIX 47
setting up connectivity for
Solaris Operating Environment,
Linux 47
Windows NT, Windows 2000,
Windows XP 42
migrating
warehouse control database 220

- mostring parameters
 - described 127
- moving and transforming data
 - warehouse tasks described 2
- Moving Average transformer
 - described 168
- MQSeries
 - creating views 176
 - described 175
 - error logging 179
 - importing messages 176
 - importing metadata 173
 - MQXMLXF error return codes
 - defined 178
 - user-defined program 178
- MQXMLXF user-defined program
 - error return codes defined 178
- multidimensional cube
 - creating a load script 205
 - creating an outline 204
 - designing a schema 204
 - loading the command script 206
 - scheduling updates 209

N

- name and address cleansing
 - error return codes defined 161
 - handling Trillium Batch System errors 160
 - importing Trillium metadata 156
 - warehouse data 153
- named tables, creating
 - Microsoft Excel data sources 54
- Network File System (NFS)
 - accessing file sources 71
- NFS (Network File System)
 - accessing file sources 71
- nickname tables, creating for DB2
 - Relational Connect sources 60
- non-DB2 relational data sources
 - defining 59
 - truncating column sizes 40
- non-DB2 sources
 - defining DB2 Relational Connect sources 60
 - defining warehouse sources and targets for Microsoft Excel 55
 - warehouse agent support 40
- non-IBM sources, supported 39
- nonparametric tests 166

O

- object definitions
 - importing 173
- Object REXX for Windows
 - enabling user-defined programs
 - Windows NT, Windows 2000, 192
- ODBC drivers
 - configuring for Informix 9.2 (without client)
 - Windows NT, Windows 2000, Windows XP 48

- odbc.ini file entry
 - example for a warehouse file source
 - AIX, Solaris Operating Environment, Linux 70
- OLAP Integration Server
 - designing a schema 204
 - running calculations from the Data Warehouse Center 224
- OLAP Integration Server metadata
 - exporting
 - Windows NT, Windows 2000, Windows XP 171
- OLAP metadata
 - importing star schemas 201
- OLAP Server database
 - loading data from a flat file 224
- OLAP Server databases
 - loading data from the Data Warehouse Center 223
- OLAP Server outline
 - updating 225
- OLE DB
 - creating views for DTS packages 198
 - creating views for table functions 198
- OLE DB Support 197
- Oracle
 - accessing as a warehouse source
 - AIX 45
 - Windows NT, Windows 2000, Windows XP 41
- ordinary delimiter
 - described 37
- outer join 114
- outlines
 - creating in the DB2 OLAP Integration Server 204
 - updating from the Data Warehouse Center 225

P

- p-value 166
 - defined 163
- parameters
 - DB2 for z/OS utility parameters
 - defined 211
 - defining user-defined program parameters 191
 - passing to a user-defined program 192
 - specifying pre-defined parameters for user-defined programs 193
 - Trillium Batch System script or JCL parameters defined 159
- password files
 - replication password files
 - described 140
- Pearson product-moment correlation coefficient 166
- period table
 - Generate Period Table transformer
 - described 149
- Pivot data transformer
 - described 151
- Point-in-Time steps
 - defining 139

- primary keys
 - described 85
 - dimension tables 116
- privileges
 - defining for DB2 DRDA sources 32
 - defining for DB2 Enterprise Server Edition sources 77
 - defining for DB2 for iSeries warehouses 74
 - defining for DB2 for z/OS warehouses 76
 - defining for DB2 UDB sources 31
 - defining for DB2 UDB warehouses 73
- process
 - described 87
 - error messages 109
- processes
 - data warehouse 2
- program groups
 - defining 190
 - described 189
- program steps
 - data warehouse 2
- programs
 - user-defined 97
 - writing user-defined programs 191
- properties
 - configuring for a data warehouse 221

R

- registry entries
 - added for the Data Warehouse Center
 - Windows NT, Windows 2000, Windows XP 253
- Regression transformer
 - described 169
- Relational Connect
 - defining source tables 61
- remote database directory
 - manipulating entries 36
- remote databases
 - cataloging for the iSeries warehouse agent 36
- remote file servers
 - accessing from the Data Warehouse Center
 - Windows NT, Windows 2000, Windows XP 66
- remote file source
 - accessing 134
 - AIX 68
 - accessing as a warehouse source
 - Solaris Operating Environment, Linux 70
 - Windows NT, Windows 2000, Windows XP 67
- remote files
 - accessing
 - Windows NT, Windows 2000, Windows XP 66
- reorg utility
 - described
 - DB2, DB2 for z/OS 91

- REORG utility
 - defining
 - DB2 UDB or DB2 for z/OS 211
- replication
 - creating replication control tables 138
 - described 135
 - password files described 140
 - setting up in the Data Warehouse Center 137
- replication programs
 - described 93
- replication source tables
 - defining 136
- replication steps
 - defining 139
 - described 88
 - using in a process 139
- right outer join 114
- rules tables
 - described 148
 - DWC 7.2 Clean Data transformer 250
- Run FTP Command File warehouse program 133
 - described 91
- run-time activities
 - monitoring steps 108
- run-time errors
 - viewing 227
- running subtotal 164
- runstats utility
 - described
 - DB2, DB2 for z/OS 91
- RUNSTATS utility
 - defining values
 - z/OS 215
 - defining values for a DB2 UDB RUNSTATS utility 215

S

- sample data
 - viewing source and target data 109
- schemas
 - adding tables and views 202
 - creating in the Data Warehouse Center 202
- security
 - Data Warehouse Center security described 20
 - defining for a warehouse 22
- select and insert SQL steps
 - incremental commit described 112
- selecting
 - data 111
 - supported DB2 data sources 25
- server mapping
 - creating for a DB2 Relational Connect source 60
- settings
 - log directory 232
- source data
 - creating a summary 118
 - filtering 117
- source tables
 - defining replication source tables 136

- sources
 - cataloging a Microsoft Access source
 - Windows NT, Windows 2000, Windows XP 51
 - cataloging a Microsoft Excel source in ODBC
 - Windows NT, Windows 2000, Windows XP 53
 - creating a join 112
 - data warehouse 2
 - DB2 types 25
 - defining DB2 sources 38
 - defining non-DB2 sources to the Data Warehouse Center 58
 - defining replication sources 136
 - described 2, 37
 - Microsoft Excel 53
 - removing a join 113
- SQL steps
 - data warehouse 2
 - defining 111
 - incremental commit described 112
 - specifying search conditions 118
 - transforming codes 114
- SQL WHERE clause
 - filtering data 117
- staging tables
 - defining replication step 139
- star joins 116
- star schema
 - description 116, 201
- starting
 - Data Warehouse Center
 - logger 9
 - server 9
- statistical transformers
 - ANOVA transformer 163
 - Calculate Statistics transformer 164
 - Calculate Subtotals transformer 164
 - chi-Square transformer 166
 - Correlation transformer 166
 - defining 163
 - described 96
 - Moving Average 168
 - Regression transformer
 - described 169
- statistics
 - defining statistical transformers 163
- step task flow
 - described 106
- steps
 - Chi-square transformer 166
 - column mapping 98
 - creating a composite summary step 118
 - defining an SQL step 111
 - defining replication 139
 - described 2, 88
 - developing 100
 - error messages 109
 - mode
 - description 100
 - development 100
 - test 100
 - monitoring run-time activities 108
 - printing to a text file 106
 - running 100

- steps (*continued*)
 - running against transient tables 101
 - SQL, selecting and inserting data 111
 - starting from outside the Data Warehouse Center 102
 - status and user-defined program feedback 194
 - testing 100, 101
- subject areas
 - described 2
- Submit z/OS JCL jobstream warehouse program 133
- subtotals
 - calculating 164
- Sybase Adaptive Server
 - accessing as a warehouse source
 - AIX 45
 - Windows NT, Windows 2000, Windows XP 41
- system tokens
 - defining for a user-defined program 191
- SYSTEM DD statement
 - example of a job step containing 158

T

- T-value 166
- table definitions
 - importing from Microsoft Access
 - Windows NT, Windows 2000, Windows XP 51
- tables
 - replication described 135
- tag language files
 - ERwin 184
 - exporting 171
 - exporting ERwin files 186
 - importing 173
 - XML 172
- target databases
 - cataloging
 - Windows NT, Windows 2000, Windows XP 52
 - cataloging for a Microsoft Excel source 55
- target tables
 - creating with DB2 Relational Connect 80
 - exporting warehouse objects from DB2 OLAP Integration Server 208
 - moving with DB2 Relational Connect 81
 - transforming 143
 - updating existing table in a remote database 82
- targets
 - data warehouse 2
 - defining 84
 - defining for a warehouse
 - Windows NT, Windows 2000, Windows XP 49
 - described 83
 - reducing the size of a target database 118
 - warehouse 84

- task flow
 - warehouse processes 107
- TCP/IP
 - configuring for z/OS 13
 - port numbers for Data Warehouse Center 254
 - updates for Data Warehouse Center 254
- test environments
 - Data Warehouse
 - steps 100
 - test mode 100
- text files
 - accessing with Copy file using FTP 65
 - accessing with FTP 64
 - loading into an OLAP Server database 224
- time
 - formats
 - changing 152
- traces
 - data
 - tracing warehouse components 228
 - files
 - accessing with CA/400 129
 - for DB2 for iSeries Load utilities 128
 - running a Data Warehouse Center component trace 230
 - viewing for DB2 for iSeries Load utilities 129
- transformer steps
 - data warehouse 2
- transformers
 - defining 88
 - defining statistical transformers 163
 - described 2, 88
 - description 96
 - error logging 231
 - FormatDate transformer 152
 - Pivot data transformer described 151
- transient tables
 - using as sources 101
- Trillium Batch System
 - error handling 160
 - JCL
 - files 157
 - parameters 159
 - script 159
 - using user-defined programs with 158
- Trillium converter
 - described 155
- Trillium DDL
 - mapping to warehouse objects 237
- Trillium geocoder
 - described 155
- Trillium matcher
 - described 155
- Trillium metadata
 - described 156
 - importing 156, 173
- Trillium parser
 - described 155

- Trillium Software System
 - cleansing name and address data 153
 - components 155

U

- UDP (user-defined program)
 - see user-defined program 189
- Update outline from file program
 - described 92
- Update outline from SQL table program
 - defined 92
- updates
 - tables in a remote database 82
- User Copy steps
 - defining 139
- user-defined program
 - program groups described 189
 - using with Trillium Batch System 158
- user-defined program steps
 - data warehouse 2
- user-defined programs
 - and step status 194
 - defined 189
 - defining 190
 - defining an agent site 190
 - defining parameters 191
 - described 2, 97
 - enabling
 - Windows NT, Windows 2000, 192
 - error return codes for MQXMLXF
 - defined 178
 - feedback 194
 - MQSeries 178
 - passing parameters 192
 - return code 194
 - specifying pre-defined parameters 193
 - writing 191
- utilities
 - DB2 for z/OS parameters defined 211
 - load and export utilities described 121

V

- verifying
 - communication 15
- views
 - creating for DTS packages 198
 - creating for MQSeries messages 176
 - creating for OLE DB table functions 198
- Visual Warehouse 5.2 DB2 UDB
 - Data Export (VWPEXPT1) program 241
 - Data Load Insert (VWPLOADI) program 242
 - Data Load Replace (VWPLOADR) 244
 - Load flat file into DB2 Enterprise Extended Edition 247
 - REORG (VWPREORG) program 246

- Visual Warehouse 5.2 DB2 UDB
 - (continued)
 - RUNSTATS (VWPSTATS) program 247
 - warehouse program file type modifiers 245
- VM file source
 - accessing as a warehouse source
 - AIX 67
 - Solaris Operating Environment, Linux 69
 - Windows NT, Windows 2000, Windows XP 65
- VWPEXPT1 program (Visual Warehouse 5.2)
 - defining 241
- VWPLDPR program (Version 5.2)
 - defining
 - AIX 247
- VWPLOADI program (Version 5.2)
 - defining 242
- VWPLOADR program (Version 5.2)
 - defining 244
- VWPREORG program (Version 5.2)
 - defining 246
- VWPSTATS program (Version 5.2)
 - defining 247

W

- warehouse administrative interface
 - adding or deleting from a warehouse 218
 - starting 8
- warehouse agent daemon
 - description 11
 - stopping
 - AIX, Solaris Operating Environment, Linux 16
 - iSeries 16
 - Windows NT, Windows 2000, Windows XP 15
 - zSeries 16
 - verifying running status
 - iSeries 12
 - zSeries
 - starting in the background 14
- warehouse agent sites
 - configuration 17
 - defining 18
 - defining for user-defined programs 190
 - description 2
- warehouse agents
 - accessing DB2 Relational Connect sources 37
 - accessing local and remote databases 35
 - adding or deleting 218
 - connecting the warehouse agent to the warehouse server 11
 - iSeries requirements for accessing remote databases 34
 - running as user processes 230
 - starting
 - on iSeries 12
 - on Windows 12

- warehouse agents *(continued)*
 - starting *(continued)*
 - on z/OS 13
 - support for DB2 sources 26
 - support for file sources 63
 - support for non-DB2 sources 40
 - switching versions 44
 - verifying status
 - iSeries 12
 - zSeries 14
 - verifying version 44
 - warehouse control database
 - backing up 217
 - description 2
 - initializing 218
 - initializing during installation 219
 - migrating 220
 - warehouse deprecated programs
 - defining a DWC 7.2 Clean Data transformer 252
 - description 94
 - rules tables 250
 - warehouse file programs 91
 - warehouse file source example 70
 - warehouse foreign keys 85
 - warehouse groups 20
 - warehouse log files 227
 - warehouse logger
 - starting and stopping 9
 - verifying status 10
 - warehouse objects
 - created by export from DB2 OLAP Integration Server 208
 - description 2
 - warehouse primary keys 85
 - warehouse privileges
 - DB2 Enterprise Server Edition databases 77
 - DB2 for iSeries 74
 - DB2 for z/OS warehouses 76
 - warehouse processes
 - description 87
 - task flow 107
 - warehouse programs
 - Copy File using FTP 133
 - DB2 for z/OS load 130
 - DB2 OLAP Integration Server programs described 92
 - description 2
 - error logging 231
 - for manipulating files 133
 - for replication 93
 - loading data into an OLAP Server database 224
 - warehouse schemas
 - adding tables and views 202
 - creating 202
 - defining 202
 - designing for DB2 OLAP Integration Server 204
 - exporting to the OLAP Integration Server 203
 - publishing metadata about 209
 - warehouse security 22
 - warehouse server
 - connectivity requirements 11
 - starting and stopping 9
 - warehouse server *(continued)*
 - verifying running status 10
 - warehouse services
 - stopping 217
 - warehouse sources
 - accessing a z/OS or VM file source 65
 - accessing local files
 - AIX 68
 - Solaris, Linux 69
 - Windows 66
 - accessing Microsoft Access databases 43
 - accessing Microsoft Excel sources 43
 - accessing remote file source 68
 - accessing remote files
 - Solaris, Linux 70
 - Windows 67
 - accessing Sybase Adaptive Server sources 41
 - accessing with DB2 Relational Connect 79
 - accessing z/OS or VM file sources 69
 - creating a join 112
 - defining
 - DB2 Relational Connect sources 60
 - DB2 sources 38
 - file sourced 63
 - Microsoft Excel sources and targets 55
 - non-DB2 sources 58, 59
 - privileges for DB2 UDB sources 31
 - source tables for DB2 Relational Connect 61
 - description 37
 - Linux 25
 - removing a join 113
 - supported versions and release levels 25
 - viewing data 109
 - warehouse steps
 - description 88
 - monitoring run-time activities 108
 - printing to a text file 106
 - testing 101
 - warehouse targets
 - cataloging for Microsoft Excel 55
 - creating a database 84
 - creating for Microsoft Excel 54
 - defining 49
 - defining DB2 for z/OS target table 76
 - description 83
 - reducing of a database 118
 - viewing data 109
 - warehouse tasks 2
 - warehouse trace information 228
 - warehouse transformers
 - ANOVA 163
 - Calculate Statistics 164
 - Clean Data 143
 - description 96
 - DWC 7.2 Clean Data 249
 - Generate Key Table 149
 - warehouse transformers *(continued)*
 - Generate Period Table 149
 - Invert Data 150
 - Pivot Data 151
 - Regression 169
 - warehouse user security 20
 - warehouse utilities
 - DB2 for iSeries Data Load Insert 125
 - DB2 for iSeries Data Load Replace 126
 - DB2 for z/OS 211
 - DB2 for z/OS RUNSTATS 215
 - DB2 UDB export 122
 - DB2 UDB load 124
 - DB2 UDB RUNSTATS 215
 - description 91
 - export with ODBC to file 122
 - load and export 121
 - modstring parameters for DB2 for iSeries Load 127
 - warehouses
 - DB2 Relational Connect 79
 - DB2 UDB for z/OS 75
 - Microsoft Access sources 50
 - warehousing
 - overview 1
 - WHERE clause
 - filtering data 117
 - work in progress
 - monitoring run-time activities 108
- ## X
- XML metadata
 - importing 176
 - XTClient syntax 103
- ## Z
- z/OS
 - configuring TCP/IP 13
 - updating environment variables 19
 - z/OS file source
 - accessing as a warehouse source
 - AIX 67
 - Solaris Operating Environment, Linux 69
 - Windows NT, Windows 2000, Windows XP 65
 - zSeries warehouse agent
 - accessing DB2 Relational Connect sources 37

Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at <http://www.ibm.com/planetwide>

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at <http://www.ibm.com/software/data/db2/udb>

This site contains the latest information on the technical library, ordering books, product downloads, newsgroups, FixPaks, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at www.ibm.com/planetwide



Program Number: 5724-E66

Printed in USA

SC27-1123-01



Spine information:



IBM® DB2 Universal Database™

Data Warehouse Center Admin Guide

Version 8.2